EXTENDED-CONWIP-KANBAN SYSTEM: CONTROL AND PERFORMANCE ANALYSIS

A Dissertation
Presented to
The Academic Faculty

by

Karin Boonlertvanich

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and System Engineering

Georgia Institute of Technology
April 2005

EXTENDED-CONWIP-KANBAN SYSTEM:
CONTROL AND PERFORMANCE ANALYSIS

Approved by:

Dr. Chen Zhou, Advisor
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Ronald Billings
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Mark Ferguson
School of Management
Georgia Institute of Technology

Dr. Paul Griffin
School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Spiridon Reveliotis
School of Industrial & Systems Engineering
Georgia Institute of Technology

Date Approved: April 6, 2005

ACKNOWLEDGEMENTS

I would like to thank my greatest source of support which is my thesis advisor, Dr. Chen Zhou. His endless intellectual support and encouragement ware a great relief.

My views on manufacturing control were greatly influenced by one of my best teacher, Dr. Paul Griffin. The inspiration of this thesis was originated from his manufacturing system course which taught me more about manufacturing control than any other sources.

I would like to acknowledge Dr. Mark Ferguson for his suggestion about including a case study for completeness of this thesis and his courtesy of providing reference papers.

I would also want to thank Dr. Spiridon Reveliotis for his comments of this dissertation and his kind collaboration when I was his TA for many times.

Thanks also go to Dr. Ronald Billings for serving as one of my thesis committee. Finally, I owe a great debt to my girlfriend, *Nok*, for helping me keep the whole thing in perspective.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURES

i, j	Unique number identifying a production stage
$\mathcal{MP}_i$	The manufacturing process of stage i
$\mathcal{B}_i$	The output buffer of stage i
$\mathcal{K}_i$	The stage-i kanban queue containing stage-i free kanbans
$\mathcal{C}$	The CONWIP queue containing free CONWIPs
$\mathcal{D}_i$	The stage-i demand queue containing demand information triggering production of stage-i
$\mathcal{DK}_i$	The authorized kanban queue containing stage-i kanbans that have been triggered by demand information from the downstream stage
$NQ(Q)$	The number of items in queue Q
k	The notation represents a kanban card
c	The notation represents a CONIWP card
d	The notation represents a demand
p	The notation represents an actual part
k_i	The number of kanbans in stage i
C	The number of CONWIPs in the system
s_i	The amount of basestock level in stage i
$J_{i,j}$	The synchronization station joining stage i and j
$\mu_i(x)$	The production rate out of stage i given that there are x parts in stage i
$A_j(n)$	The production trigger for stage j in period n

$PO_j(n)$ The production objective for stage j in period n

P_j^{min} The minimum production capacity of stage j

P_j^{max} The maximum production capacity of stage j

$I_j(n)$ The amount of inventory held in the stage j output buffer in period n

$D(n)$ The actual demand quantity in period n

$\tilde{D}(n)$ The estimated demand quantity in period n

$\{I_j(n-1)\}$ The set of inventories held in the output buffers of the immediate upstream stages of stage j in period n

$D_j(n)$ The number of demands held at stage j in period n

SUMMARY

We analyzed and compared various pull type control policies and conducted comprehensive simulation comparison studies. The important insights include an understanding of the potential impacts and expected effectiveness of different control mechanisms. Based on the analysis, we designed a new class of control policy called Extended CONWIP Kanban control (ECK). Simulation results showed that the new control policy can achieve higher service levels with lower inventories than other policies including hybrid ones.

We developed a state space comparison methodology to compare various policies with that of optimum solution. Our analysis reveals that ECK policy provides the most adjustment flexibility toward the optimum. We also proposed a parameter design procedure and an adaptive parameter adjusting system. A case study verifies the advantages and utilities of the ECK policy. The effects of demand uncertainties are also examined. Beside the cost saving benefits, another major advantage of the ECK policy is its robustness. That is, the total cost of the ECK policy is less sensitive to demand or system variations. Therefore, this research contributes a new control policy and understanding of control mechanisms, and provides valuable insights to production managers.

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

Flow-line manufacturing systems represent the most prevalent process structure in industry for production of discrete items. Flow line system can be divided into stages. Each stage may be seen as a production/inventory system consisting of a manufacturing process and an output buffer. A major challenge in designing and operating flow lines is to achieve high customer service level while stay lean.

Determination of the mechanism to control the flow of materials through the manufacturing system is one of the most important decisions. Material flow control is to addresses the problems of when and how much to authorize parts to be processed at each stage in order to achieve a specified customer service level, while minimizes work-in-process. Difficulties in the control arise due to production and demand variabilities.

One approach to deal with material flow control problem is to formulate a stochastic optimal control problem and then try to determine an optimal control policy (Gershwin 1994). Thus far, this approach has been successful only for very simple systems. Moreover, an optimal policy for a realistic system is likely to be too complicated to be implemented.

A more practical approach is to find a simple sub-optimal policy that is easy to be implemented. Much of the research effort in this area has focused on simple control systems that depend on a small number of parameters per stage and have often emerged from actual industrial practice (see Hopp and Spearman (1996), Veatch and Wein (1994) and Liberopoulos and Dallery (2000) as an example). From the 80s, Japanese Just-In-Time (JIT) manufacturing approach has triggered various 'pull production systems' that emphasize the importance of production control that react to actual demand rather than future demand forecasts.

Push systems schedule periodic releases of raw materials into the production line, while pull systems authorize parts to be processed in response to the actual demand arrival. Push systems batch and control release rate (and hence throughput) and observe work-in-process (WIP) from time to time, while pull systems control WIP and observe throughput. See Spearman *et al.* (1990), Spearman and Zazanis (1993), Hopp and Spearman (1996) for advantages of the pull systems over the push systems.

A pull mechanism can be implemented in many ways. The best known is a Kanban policy (Monden 1983, Ohno 1988, Shingo 1989, Rees *et al*. 1987, Philipoom *et al*. 1987, and Berkley 1992). The Kanban control was originally used in Toyota production lines in the mid-seventies and is often considered to be closely associated with the philosophy of the JIT approach (Zipkin 1991 and Groenvelt 1993). In the Kanban control system, Production authorization cards, called Kanban, are used to control and limit the releases of parts into each production stage. The advantage of this mechanism is that the number of parts in every stage is limited by the number of kanbans of that stage.

Its disadvantage is that the system, especially in the upstream stages, may not respond quickly enough to changes in the demand.

Another pull control system originated from inventory control technique is Base Stock policy (Clark and Scarf 1960 and Kimball 1988). The Base Stock system was initially proposed for production/inventory systems with infinite production capacity and uses the idea of a safety stock for finished good inventory as well as safety buffers between stages for coordination. In the Base Stock control system every stage has a target inventory of finished parts, called basestock. When a demand for an end item arrives, it is immediately transmitted to every stage to authorize the release of a new part. An advantage of this mechanism over JIT is that it avoids demand information blockage by transferring the demand information immediately to all production stages. The down side is that it provides no limit on the number of parts in the system.

Constant Work-In-Process (CONWIP) control system proposed by Spearman *et al.*(1990) uses a single card type to control the total amount of WIP permitted in the entire line. It can also be viewed as a single stage Kanban system. In effect, the CONWIP control system can be considered as a pull system at the end of the line, or a push system from the beginning of the line towards the end. The pushing part of the system can suffer with problems associated with a traditional push system.

Since the basestock mechanism quicker reacts to demand and the Kanban mechanism achieves better coordination and limits in work-in-process inventories, intuitively, combining respective merits of Base Stock and Kanban control mechanisms may entail many potential benefits. Buzacott (1989) proposed a hybrid control system, called Generalized Kanban control system, which includes the Kanban and Base Stock control system as special cases. However, it is more versatile but also more complex than the Base Stock and Kanban controls. The complexity is due to that the demand information flow is relayed upstream rather than directly transfered upon arrival. The Generalized Kanban control system depends on two parameters per stage, (i) the amount of base stock of finished parts, and (ii) the number of kanbans.

Recently, Dallery and Liberopoulos (2000) introduced a new pull type control mechanism called Extended Kanban control system which is also a mixture of Base Stock and Kanban systems. This mechanism is conceptually less complicated than Generalized Kanban control system, since the demand information is now directly transferred to every stage using global demand flow as in the Base Stock system. In addition, unlike Generalized Kanban system, the roles of basestock and kanban are completely separated due to the global demand flow. Thus, it is potentially easier to be implemented. However, one drawback of Extended Kanban compared with Generalized Kanban is that it requires the amount of kanbans to be at least as large as the basestock level, which limits its configuration flexibility. The Extended Kanban control system also includes both the Base Stock and Kanban systems as special cases

Another hybrid control system is CONWIP Kanban policy proposed by Bonvik (1997). This control system combines local work-in-process control mechanism using kanbans and global inventory control using CONWIPs. Demand information is transferred to the first stage using CONWIP mechanism. However, this information flow can still be halted if there is no finished goods inventory at the time demand arrives. It has been shown that this hybrid system could achieve better performance than using Kanban, Base Stock, or CONWIP controls alone.

Although much works in the literature has been done on individual control systems, few comprehensive comparison studies exist. This is partly due to the fact that different systems have been described within different frameworks. This thesis presents a unified comprehensive comparison of pull control policies in multi-stage manufacturing systems. We show differences and similarities of the control actions in each control policy. Three traditional pull control mechanisms, which are CONWIP, Base Stock, and Kanban, are considered. Superposition or hybrids of the traditional mechanisms are also included. They are CONWIP-Kanban, Generalized Kanban, and Extended Kanban control systems. A new class of hybrid control called Extended CONWIP Kanban control is also developed and it performance is compared with the aforementioned controls.

So far, most research has focused on optimizing parameters of a specific control policy under some given constraints or objectives. Since the 90s, the internet has lead to great product proliferation and shorter life. The combination resulted in more systematic fluctuations in demand. To respond to these changes, control parameters should also be adjusted accordingly. Moreover, even though the demand driven pull system is not prone to probabilistic forecasting errors, the control parameters should be adjusted to respond to changes in demand process. Only a few papers address the issue of how to adjust the system over time (Tardif and Maaseidvaag 1999 and Takahashi 2003).

This thesis is organized as follow. First, we study the characteristics and dynamics of various types of control policies including the hybrid policies in Chapter 2. In Chapter 3, we investigate a new hybrid control policy called Extended CONWIP Kanban policy (ECK), which is the superposition of Base Stock, CONWIP, and Kanban control system, see Figure 1-1. We then show the advantages of ECK control policy via simulation studies in Chapter 4 and via state-space analysis in Chapter 5. We purpose a method on designing the parameters of ECK policy in Chapter 6 and a method on adjusting control parameters according to changes in demand in Chapter 7. A case study abstracted from an iron and steel production plan is presented in Chapter 8. Finally, conclusion, contributions and future research are provided in Chapter 9.

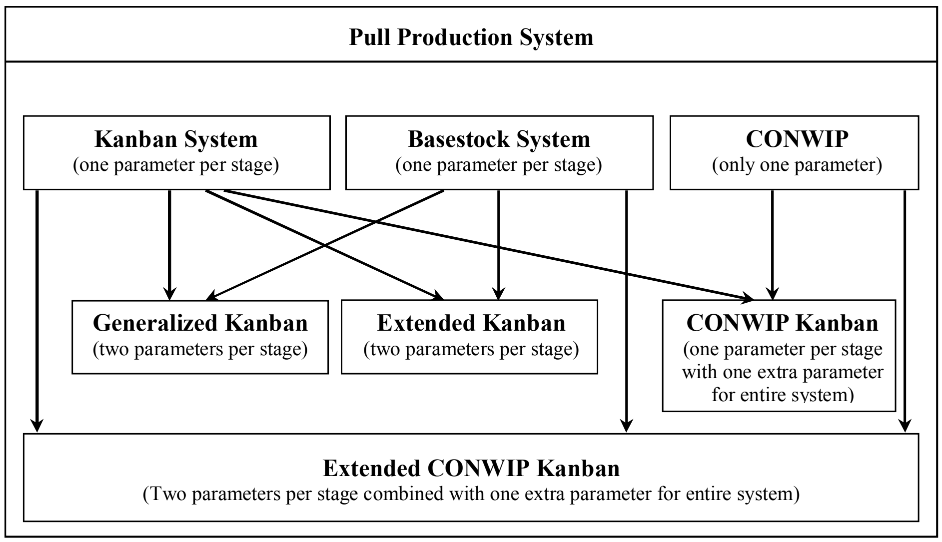


Figure 1-1: Pull Production Systems

1.2 Literature Review

The simplest control policy for a production system is to run the production at its maximum capacity and speed, assuming an unlimited demand and stock limit. The control parameters of this policy, even if they are not explicit, are the capacities of buffers between stages along the production line. A control policy is simply: producing whenever there are parts to process and a place to put the finished product. There are several variations of this control policy, such as blocking before service and blocking after service. See Dallery and Gershwin (1992) for a review.

There are enormous amount of literatures on control policies for manufacturing systems; however, we will consider only policies that use the movement of tokens or authorization cards in the manufacturing system to authorize the part(s) to be released or processed.

For Kanban policy, Rees *et al.* (1987) applied the Toyota approach under fluctuating product mix by using the next period's forecasted demand and the last period's observed lead time. Using lead time and forecasted demand information, they estimated the density function of lead times as well as the probability mass function (*pmf*) of n, the number of kanbans. From the estimated pmf, they determine the number of kanbans that minimize shortage and holding costs.

Philipoom *et al.* (1987) used a simulation to determine lead times at work stations and thereby the number of kanbans required at each station to prevent backorders in a dynamic production environment. They also described factors that influence the number of kanbans required in implementing JIT production techniques. These include throughput velocity, process variation, machine utilization, and autocorrelation of processing times.

Queueing theory has also been used by several researchers to determine the number of kanbans required in stochastic production systems. Deleersnyder *et al.* (1989) determined the appropriate number of kanbans for a manufacturing system with stochastic demands and machine failures by developing a discrete time Markov model of a single card Kanban system.

CONWIP control system was proposed by Spearman *et al.* (1991). This policy fixes a population constraint for the system, and when this constraint is reached, orders are only released to the system in response to actual customer demands for finished parts at the finished good inventory. Once a raw part is released at the input of the system, it is pushed through the system as fast as possible.

For the Base Stock policy, initial inventories in each output buffer, called basestocks, control how much material is held in the line when waiting for another demand. It can be shown that this is an optimal control policy for an uncapacitated manufacturing system (Clark and Scarf 1960). However, in a two-machine line with finite capacities and unreliable machines, Veatch and Wein (1994) demonstrated that the choice between a basestock and a Kanban control depends on the location of the system bottleneck. If the upstream machine is slower, Base Stock control is preferred, otherwise Kanban control is better. This seems to be due to the different information flows in these two control disciplines: global information flow in Base Stock control versus local information flow in Kanban control.

Recently, several authors have proposed control schemes that are combinations and supersets of the control policies mentioned above. One general scheme is the Generalized Kanban control proposed by Buzacott (1989). It contains Kanban and Base Stock control policies as special cases. Generalized Kanban control system depends on two parameters per stage, the basestock level of finished parts and the number of kanbans. Similar framework called Extended Kanban control policy has recently been proposed by Frein, Di Mascolo, and Dallery (1994) and Dallery and Liberopoulos (1995). The Extended Kanban control policy, like the Generalized Kanban control, depends on two parameters per stage, the basestock and the number of kanbans, but its operation is simpler than that of the Generalized Kanban control. Another difference is that the production capacity of the Extended Kanban control system depends only on the number of kanbans at each stage, while the production capacity of the Generalized Kanban control system depends on both the number of kanbans and the basestock level at each stage.

For the adjustment of the amount of kanbans over time, Tardif and Maaseidvaag (1999) proposed an adaptive control mechanism for Kanban control system. The control mechanism releases or retrieves extra Kanban cards from the system according to the finished goods inventory and backorder level. It has been shown that this adaptive Kanban control outperformed original Kanban control for the systems having Poisson demand arrival and exponential service times.

Takahashi (2003) proposed a reactive control mechanism for Kanban control system. The system adjusted the amount of Kanban cards according to a detected change in demand process using the time series data of the finished goods inventory level.

CHAPTER 2

PRODUCTION CONTROL POLICIES

In this chapter, we will describe the operation and control characteristics of the control systems found in the literature. We will proceed from traditional to hybrid control policies. Traditional policies include CONWIP, Base stock and Kanban policies. The simplest is CONWIP control, which has only single control parameter for the entire production line. Then, we discuss Base Stock and Kanban controls which requires one control parameter for each production stage. Hybrid policies include CONWIP-Kanban, Generalized Kanban and Extended Kanban policies. The first and simplest hybrid control presented here is the CONWIP-Kanban control which requires one control parameter for each stage and one additional parameter for the entire production line. We then discuss Generalized Kanban and Extended Kanban controls which requires two control parameters for each production stage.

2.1 Modeling convention

In this thesis we consider a type of manufacturing system in which the production of parts proceeds in stages. Each stage is a production/inventory system made up of a manufacturing process and an output buffer. The manufacturing process may consist of a single machine or a subnetwork of several machines.

There are many reasons for aggregating production activities into stages and control material flow only in between stages. First, in most practical situations, production activities are naturally grouped into well identifiable production stages which operate independently from one another. Second, once a part has entered a production stage, continuous processing on that part might be required and controlling the production at each individual machine may not be appropriate in such cases. Finally, aggregating production activities into stages results in fewer control points and makes the control problem simpler and easier to implement.

We restricted our attention to manufacturing systems having stages in series and products go through the same sequence of stages without any re-entrants or defects. The first manufacturing stage is fed by raw parts which are assumed to be infinite. The subsequence stages are fed by part in the output buffer of its upstream stage. We also assume that there is only a single part type and no machine setup times.

Figure 2-1 shows a manufacturing system with two stages in tandem. Each stage consists of two machines. The manufacturing process at each stage is drawn as an oval, machines in each manufacturing process are drawn as circles and the intermediate and output buffers are drawn as triangles. This is compliance with standards developed by the American Society of Mechanical Engineers (ASME) and Japanese industry standards (JIS-Z-8280), see also Bonvik (1996).

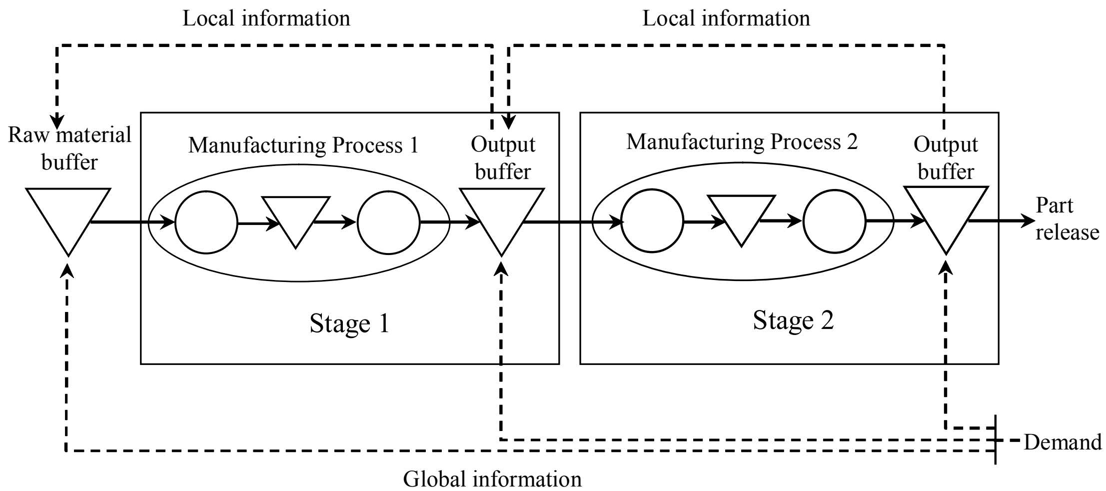


Figure 2-1: A manufacturing system having two stages in tandem

The information flow in the system can be divided into two classes: global information flows and local information flows. Global flows connect individual stages in the production system to the demand process, without going through intermediate production stages. Local information flows connect production stages to the adjacent buffer inventories. We use the graphical convention that dashed lines indicate information flows which can be demand information or production authorizations.

To show the control dynamics of each control policy, we used common modeling approach which modeled different type of control policies as a queueing network with synchronization stations. A synchronization station consists of a server with instant service time, fed by two or more queues. As soon as there is at least one item in each of the queues that feed the synchronization server, these items instantaneously move through the station. Items or parts that enter the synchronization stations may be separated into more or joined into fewer items upon exiting the station.

Solid lines represent material flows. A lower case letter on each line represents the contents passing along that line. The meaning of symbols used in this chapter can be found in the nomenclatures. For the sake of simplicity, the queueing network models of each control policy are shown in a two-stage production system format. Multi-stage system can be modeled based on these two-stage models.

2.2 CONWIP Control System (CW)

CONWIP mechanism maintains a WIP level upper bound for the entire system. When the preset WIP level is reached, no new jobs are authorized for release to the system before some job leaves. This occurs in response to demand events. A CONWIP line can be seen as controlled by a single kanban cell encompassing all stages.

Figure 2-2 shows the queueing network model of a single-product CONWIP control policy having two manufacturing stages in tandem. Even though there are two stages drawn here, CONWIP production control is executed only at the entry of the manufacturing system and the intermediate buffer, $\mathcal{B}_1$, plays no control action. CONWIP control is indeed considered as a single-stage control. We depicted the system as two stages for the consistency purpose, since all other control policies will be described later are depicted as two manufacturing stages.

$\mathcal{MP}_1$ and $\mathcal{MP}_2$ represent manufacturing stage 1 and 2, while $\mathcal{B}_0$ represents raw material buffer. Queue $\mathcal{B}_i$ is the output buffers of stage i. Queue $\mathcal{MP}_i$ is the total amount of parts that has been released to stage i. Queue $\mathcal{D}$ contains the demand and queue $\mathcal{C}$ contains CONWIP cards/signals.

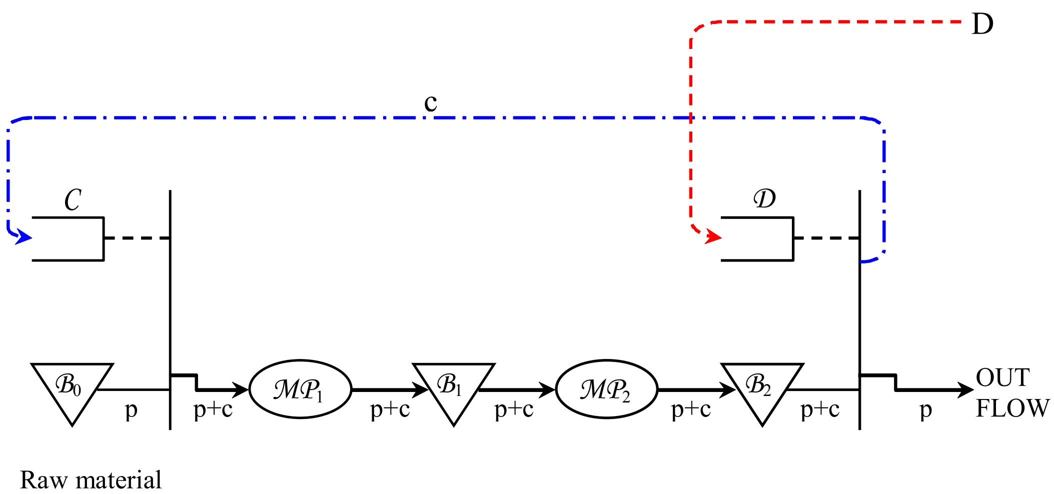


Figure 2-2: A two-stage production line controlled by CONWIP

When the system is in its initial state, that is before any demands have arrived at the system, buffer $\mathcal{B}_1$ contains no part. Only buffer $\mathcal{B}_2$ contains C finished parts attached with the CONWIP cards.

The CONWIP policy operates as follows. When a customer demand arrives at the system it requests the release of a finished product from $\mathcal{B}_2$ to the customer. At this time there are two possibilities:

- If a part is available in $\mathcal{B}_2$ (which is initially the case), it is released immediately to the customer and the CONWIP card is detached from the part and transferred to queue $\mathcal{C}$.
- Otherwise, the demand is backordered and waits in $\mathcal{D}$ until a new part completes from the upstream stage arrives.

For other stages beside the last stage, they will operate in the same way as push system, i.e. parts move downstream without any blocking.

The CONWIP control is a very simple control mechanism that depends only on one parameter for the entire system, the amount of CONWIP, C. It influences both the transfer of finished parts downstream and the transfer of demands upstream through the system. There is no demand transfer between each stage except the last and the first stage.

The production capacity or the maximum production rate of the system is affected only by the amount of CONWIP card, C. The total amount of parts in the system is bound by C and can be expressed as follow.

$$NQ(\mathcal{C}) + \Sigma\, NQ(\mathcal{MP}_i) + \Sigma\, NQ(\mathcal{B}_i) = C, \qquad i = 1,\ldots, N \qquad (2.1)$$

If a stage fails in a CONWIP line, the amount of material downstream of it will be gradually flushed out of the system by the demand process. These demand events will trigger the release of new raw parts into the system. When all CONWIP cards accumulate in front of the failed machine, the release of new jobs to the system will then stop.

CONWIP can be implemented by associating a single card with each part, authorizing its presence in the system. Whenever a part leaves the finished goods inventory, its card is detached and sent to the first production stage, authorizing another part to enter the system. All other stages always authorized to work on any part released to the system, so passing card to these machines is not necessary.

2.3 Base Stock Control System (BS)

Base Stock control is a simple pull control mechanism for coordinating multi-stage production system where the term "base stock" is borrowed from inventory control theory. It tries to maintain a certain amount of finished parts in each output buffers, subtracting backlogged finished goods demand, if any. This amount is called the *basestock level* of each stage.

To operate a Base Stock control, it is necessary to transmit demand information to all production stages as demand occurs, which can be called global demand information. This can be done by using either card-based system or computer-based system.

Figure 2-3 shows the queueing network model of a single-product Base Stock control policy having two stages in tandem. Queue $\mathcal{D}_i$ contains the demand. In the Base Stock control, there is no coordination between consecutive stages, i.e. a part is authorized to be transferred downstream by the global demand signals.

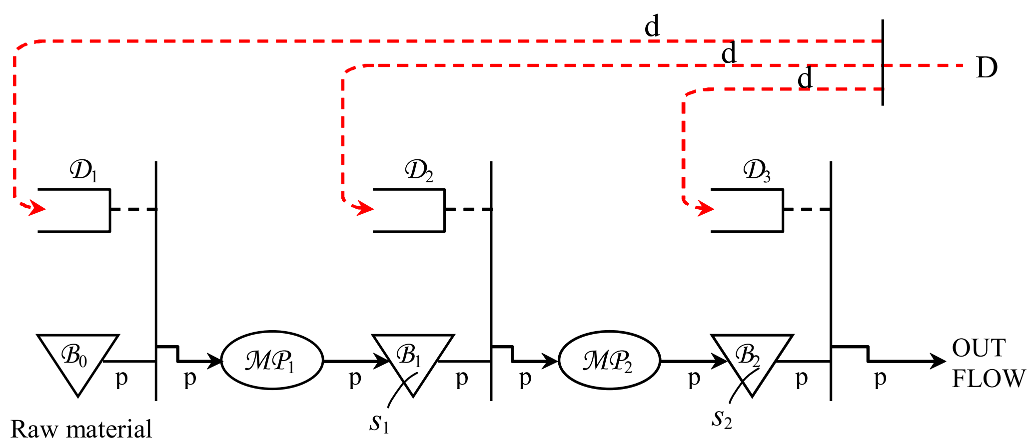


Figure 2-3: A two-stage production line controlled by Base Stock control

When the system is in its initial state, that is before any demands have arrived at the system, $\mathcal{B}_i$ contains s_i stage-i basestock level of finished parts. The Base Stock policy operates as follows. When a customer demand arrives at the system it is separated into

N+1 demands, each one is immediately transferred its respective queue $\mathcal{D}_i$ and last one joins queue $\mathcal{D}$ requesting the release of a finished product from $\mathcal{B}_2$ to the customer. At this time there are two possibilities:

- If a part is available in $\mathcal{B}_i$, it is released immediately to the downstream stage and produce one to make up the basestock or to the customer for the last stage and this demand d_i is satisfied.
- If no part is available in $\mathcal{B}_i$, the demand is backordered and waits in queue $\mathcal{D}_i$ until a new part completes from the upstream stage arrives.

Base Stock control is a simple control mechanism that depends only on one parameter per stage, namely s_i, $i = 1,\dots, N$. This parameter influences the transfer of finished parts downstream through the system, but it does not interfere the transfer of demand information upstream.

The production capacity of the system does not depend on s_i, actually, it is given by the production capacity of the bottleneck stage. The finished part buffers are bound by basestock level; however, the WIP levels in each stage are unbounded. When a stage fails, the demand process will continue to remove parts form the output buffer, and the machines downstream of the failure will operate normally until they become starved of parts to process. The upstream stages continue to receive direct demand information and will operate and release parts as usual. There will therefore be an unbounded build-up of inventory in front of the failed machine.

The relationship between each queue in the Base Stock system is shown in Equation 2.2 and the bound on the output buffers is shown in Inequality 2.3.

$$NQ(\mathcal{D}_i) + NQ(\mathcal{MP}_i) + NQ(\mathcal{B}_i) = s_i + NQ(\mathcal{D}_{i+1}) \qquad i = 1,\ldots, N \qquad (2.2)$$

$$NQ(\mathcal{B}_i) \leq s_i, \qquad i = 1,\ldots, N \qquad (2.3)$$

2.4 Kanban Control System (KB)

Kanban control system is probably the most famous pull-type mechanism for multi-stage production system during the last few decades. This control discipline limits the amount of inventory to a fixed maximum for each cell consisting of a stage and its output buffer, where the maximum is equal to the number of kanban circulating within the cell.

There are many versions of Kanban control, e.g. single-card Kanban, two-card Kanban, etc. A comparison of different kanban systems can be found in Muckstadt and Tayur (1995). In this thesis, we consider only a simple Kanban system, which is coincides with that of Buzacott and Shanthikumar (1993).

Figure 2-4 shows the queueing network model of a single-product simple Kanban control policy having two stages in tandem. Queue $\mathcal{B}_i$ is the output buffers of stage i containing both finished parts and stage-i kanbans. Queue $\mathcal{K}_i$ contains stage-i kanbans. The kanban movement is shown by the blue dashed-dotted line.

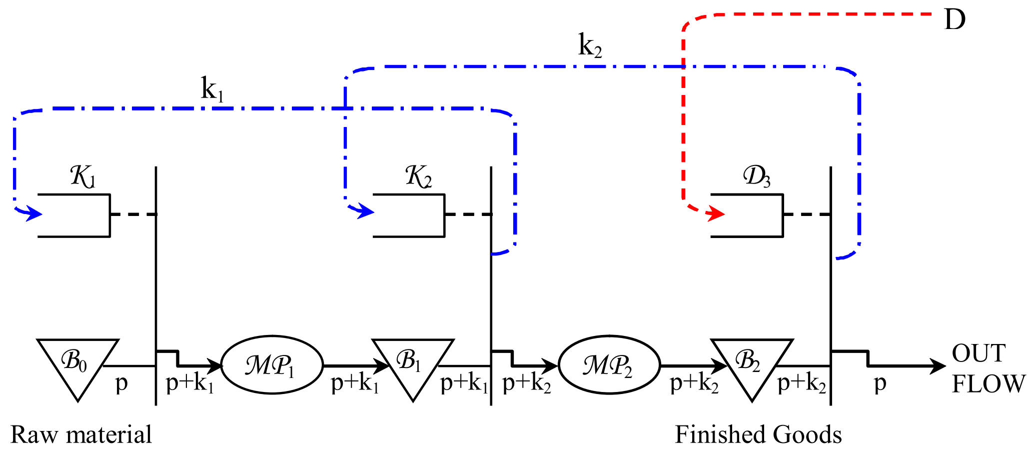


Figure 2-4: A two-stage production line controlled by Kanban

When the system is in its initial state, $\mathcal{B}_i$ contains k_i stage-i finished parts, each part having a stage-i kanban attached to it, and all other queue are empty. The Kanban control operates as follows. When a customer demand arrives at the system it joins queue $\mathcal{D}$ requesting the release of a finished product from $\mathcal{B}_2$ to the customer. At that time there are two possibilities:

- If a part is available in $\mathcal{B}_2$ (which is initially the case), it is released to the customer after liberating the stage-2 kanban that was attached to it. This kanban is transferred upstream to $\mathcal{K}_2$ carrying with it a demand signal for the production of a new stage-2 finished part.
- If no part is available in $\mathcal{B}_2$, the demand is backordered and waits in $\mathcal{D}$ until a new part completes from stage 2 arrives in $\mathcal{B}_2$. The newly finished part will be released to the customer instantly and the attached kanban will transfer to $\mathcal{K}_2$ queue instantly too.

As soon as a kanban signal arrives in $\mathcal{K}_2$, it authorizes the production of a new part in stage-2. Again at this time two things may happen:

- If a part to which a stage-1 kanban is attached is available in $\mathcal{B}_1$, it is instantaneously detached stage-1 kanban and attached stage-2 kanban. At the same time, the pair (part, stage-2 kanban) is released in to $\mathcal{MP}_2$. The stage-1 kanban is transferred upstream to $\mathcal{K}_1$ authorizing the release of a raw part from $\mathcal{P}_0$ into $\mathcal{MP}_1$.
- If no part is available in $\mathcal{B}_1$, stage-2 kanban waits in $\mathcal{K}_2$ until a newly finished part arrives in $\mathcal{B}_1$.

This way the customer demand information is transferred upstream by kanban signal. If at some stage i a finished part is not available in $\mathcal{B}_i$, no kanban is transferred upstream and the demand information is temporarily stopped; it is resumed when a part becomes available again in $\mathcal{B}_i$. Thus, the philosophy of the kanban control is that a customer demand is transmitted upstream from stage i only when a finished part is released downstream from stage i.

The kanban control is a simple control mechanism that depends only on one parameter per stage, namely k_i, $i = 1,\ldots, N$. These parameters influence both the transfer of finished parts downstream through the system and the transfer of demands upstream through the system.

In a Kanban system the transfer of a finished part form $\mathcal{B}_i$ into $\mathcal{MP}_i$ is totally synchronized with the transfer of a demand from $\mathcal{K}_{i+1}$ (or $\mathcal{D}$ if $i = N$) into $\mathcal{K}_i$. The invariant of kanban mechanism of each stage can be expressed as follow.

$$NQ(\mathcal{K}_i) + NQ(\mathcal{MP}_i) + NQ(\mathcal{B}_i) = k_i, \qquad i = 1,\ldots, N \tag{2.4}$$

This also implies that both the WIP and the number of finished parts in each stage i are bounded by k_i.

2.5 CONWIP Kanban Control System

In some cases, the local inventory build-up in Base Stock and CONWIP control is excessive. For example, if some stage is a bottleneck stage, any inventory build-up in front of it will remain in the system for a long time. If the upstream stages of that bottleneck stage are relatively fast and reliable, we may choose to limit the inventory build-up before the maximal level is reached. We therefore also investigate system that is hybrid of Kanban and CONWIP control, that is, where demand information is propagated directly from the finished-product buffer to authorize the production in the first stage via CONWIP mechanism and there also are inventory limits in each stage as in Kanban control.

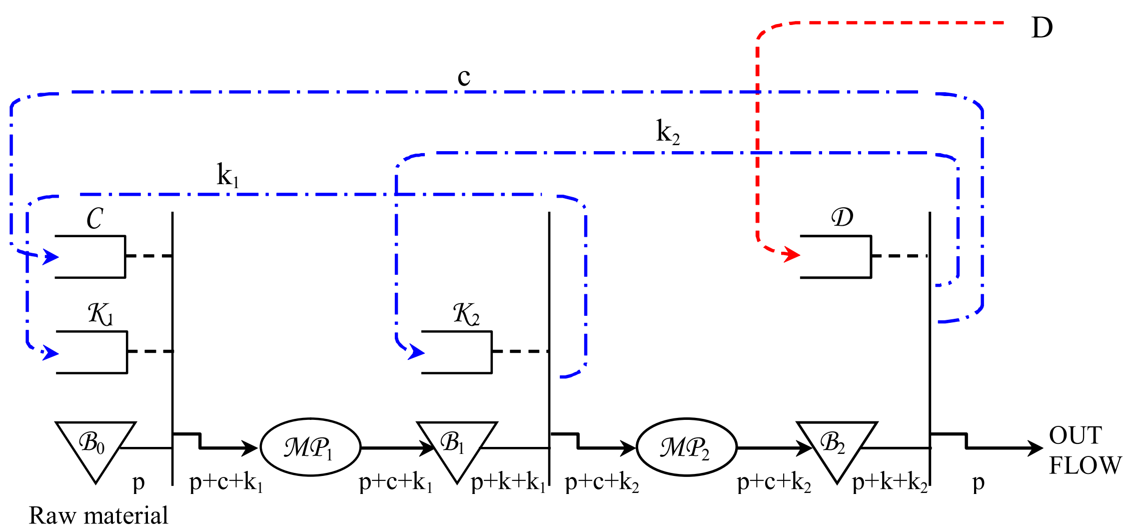


Figure 2-5: A two-stage production line controlled by CONWIP Kanban

Figure 2-5 shows the queueing network model of a single-product CONWIP Kanban control policy having two stages in tandem. Queue $\mathcal{B}_i$ is the output buffers of stage i containing finished parts, stage-i kanbans and CONWIP cards. Queue $\mathcal{K}_i$ contains stage-i kanbans and queue $\mathcal{C}$ contains CONWIP cards. The kanban and CONWIP card movements are shown by the blue dashed-dotted line.

When the system is in its initial state, queue $\mathcal{B}_2$ contains k_2 stage-2 finished parts, each part having a stage-2 kanban and a CONWIP card attached to it. Queue $\mathcal{B}_1$ contains C - k_2 stage-2 finished parts, each part having a stage-2 kanban and a CONWIP card attached to it There are $k_1 - (C\text{-}k_2)$ free stage-1 kanbans in queue $\mathcal{K}_1$. All other queues are empty. We set the assumption that the sum of all kanbans is greater than or equal to the CONWIP level.

CONWIP Kanban control operates as follows. When a customer demand arrives at the system it joins queue $\mathcal{D}$ requesting the release of a finished product from $\mathcal{B}_2$ to the customer. At that time there are two possibilities:

- If a part is available in $\mathcal{B}_2$ (which is initially the case), it is released to the customer after liberating the stage-2 kanban and the CONWIP card that was attached to it. This kanban is transferred upstream to $\mathcal{K}_2$ carrying with it a demand signal for the production of a new stage-2 finished part. The CONWIP card will be transferred to queue $\mathcal{C}$ to authorize release of raw material.
- If no part is available in $\mathcal{B}_2$, the demand is backordered and waits in queue $\mathcal{D}$ until a new part completes from stage 2 arrives in $\mathcal{B}_2$. The newly finished part will be

released to the customer instantly and the attached kanban will transfer to $\mathcal{K}_2$ queue instantly too.

As soon as a kanban signal arrives in $\mathcal{K}_2$, it authorizes the production of a new part in stage-2. Again at this time two things may happen:

- If a part to which a stage-1 kanban is attached is available in $\mathcal{B}_1$, it is instantaneously detached stage-1 kanban and attached stage-2 kanban. At the same time, the pair (part, stage-2 kanban) is released in to $\mathcal{MP}_2$. The stage-1 kanban is transferred upstream to $\mathcal{K}_1$ authorizing the release of part into stage 1.
- If no part is available in $\mathcal{B}_1$, stage-2 kanban waits in $\mathcal{K}_2$ until a newly finished part arrives in $\mathcal{B}_1$.

For one exceptional case, when either a kanban signal or a CONWIP signal arrives at queue $\mathcal{K}_1$ for kanban or queue $\mathcal{C}$ for CONWIP, it needs to wait for the other signal in order to release the raw material into stage 1.

This way the customer demand information is transferred upstream by kanban signal and transferred to the first stage by the CONWIP signal. If at some stage i a finished part is not available in $\mathcal{B}_i$, no kanban is transferred upstream and the authorization for releasing a part upstream is temporarily stopped; it is resumed when a part becomes available again in $\mathcal{B}_i$. Moreover, this CONWIP control will limit the total WIP in the system because even if there is a free kanban at the first stage, the raw material will not be release unless the total work-in-process of the entire system is below the CONWIP limit.

CONWIP kanban control is a hybrid control mechanism that depends only on one parameter per stage, namely k_i, i = 1,…, N and one additional parameter for the entire system, C. The number of kanbans and CONWIPs influence both the transfer of finished parts downstream through the system and the transfer of demands upstream through the system.

2.6 Generalized Kanban Control System (GK)

The Generalized Kanban Control System (GK) is a modified version of the Kanban control. Figure 2-6 shows the queueing network model of a Generalized Kanban control system having 2 stages in series.

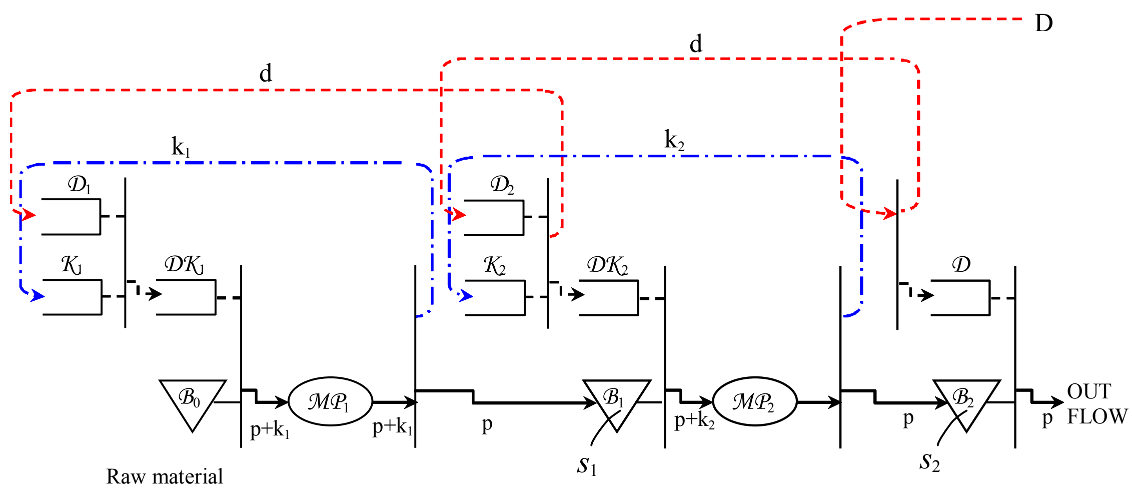


Figure 2-6: A two-stage production line controlled by Generalized Kanban control

In Generalized Kanban control each stage i has k_i kanbans to authorize the production of stage i. Initially, all kanbans, k_i, in stage i are stored in queue $\mathcal{K}_i$. Buffer $\mathcal{B}_i$, $i = 1,\ldots, N$, has s_i finished parts of stage i with $\mathcal{B}_0$ represents the raw material buffer. The demands of the production of stage-i parts are now stored in two queues: queue $\mathcal{D}_i$ only contains demands, where as queue $\mathcal{DK}_i$ contains stage-i kanbans that have been triggered by demand information from the downstream stage.

As opposed to the Kanban control, the Generalized Kanban control depends on two parameters per stage, which are the amount of kanbans in each stage, k_i and the basestock level of that stage, s_i.

Generalized Kanban control operates as follows. When a customer demand arrives at the system it is instantaneously split into two demands: the first demand will join queue $\mathcal{D}$ requesting the release of a finished product from $\mathcal{B}_2$ to the customer, the second demand will join queue $\mathcal{D}_2$ requesting the production of stage 2:

When the first demand arrives at $\mathcal{D}$:

- If a part is available in $\mathcal{B}_2$ (which is initially the case), it is released to the customer.
- Otherwise the demand is backordered and has to wait for a finished product to arrive in $\mathcal{B}_2$.

When the second demand arrives at $\mathcal{D}_2$:

- If a stage-2 kanban is available in $\mathcal{K}_2$ (which is initially the case), a demand information is immediately transmitted upstream to $\mathcal{D}_1$. Stage-2 kanban will move to queue $\mathcal{DK}_2$ authorizing the production of stage 2.
    - If a new part is available in $\mathcal{B}_1$, it is instantaneously merged with stage-2 kanban in $\mathcal{DK}_2$ and the pair (part and kanban) is released into $\mathcal{MP}_2$.
    - Otherwise the kanban has to wait in queue $\mathcal{DK}_2$ for a finished part to arrive at $\mathcal{B}_2$.

- If no stage-2 kanban is available in $\mathcal{K}_2$, the demand has to wait for a stage-2 kanban. This demand information will be stopped going upstream.

There are three major differences between the Generalized Kanban and the simple Kanban controls:

The first difference is that the demand move upstream separately from the release of parts downstream. In a Generalized Kanban control, for a demand to be transferred upstream of a given stage i, only two conditions must be met: (1) there must be a demand in the queue $\mathcal{D}_{i+1}$ and (2) there must be a stage-i + 1 kanban in $\mathcal{K}_{i+1}$. In a kanban control, demands always move together with kanbans, therefore, in addition to those previous two conditions, a demand will be transferred to the upstream stage when there is a finished part in the output buffer of stage i. This implies more decoupling in the return of demands and kanbans in a Generalized Kanban control than in a Kanban control. This is also implies that the transfer of a demand upstream to a given stage is not completely synchronized with the transfer of a finished part to the next stage.

The second difference between the Generalized Kanban control and the Kanban control is that in a Kanban control, kanban is detached from a part only when the finished part of a stage i to which the kanban is attached is transferred from $\mathcal{B}_i$ to stage i+1. In a Generalized Kanban control, a kanban is detached as soon as the finished part completes its manufacturing process in $\mathcal{MP}_i$. This implies that the kanbans are freed earlier in a Generalized Kanban control than in a Kanban control.

The third difference is that in a Generalized Kanban control, there are initially, for each stage i, s_i finished parts without kanbans attached to them in the output buffers and k_i stage-i free kanbans in queue $\mathcal{K}_i$. This allows for a partial decoupling of the transfer of parts downstream and the transfer of demands upstream. In the special case where $k_i = s_i$ for all stages i, $i = 1,\ldots, N$, the Generalized Kanban control and the Kanban control have been shown to be equivalent.

2.7 Extended Kanban Control System (EK)

The Extended Kanban Control System is similar to Generalized Kanban control just discussed. It differs from the GK in that the demand information are not relayed through stages, as shown in Figure 2-7.

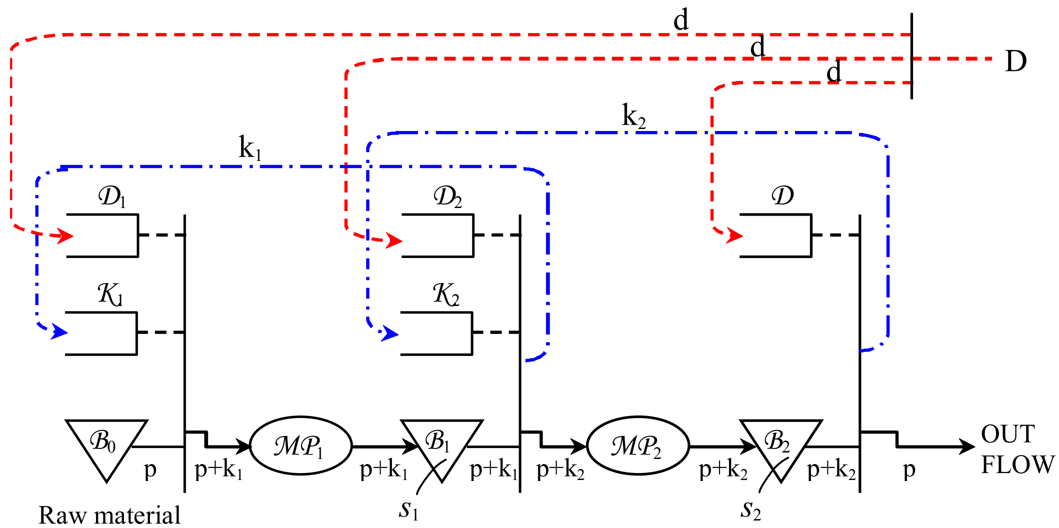


Figure 2-7: A two-stage production line controlled by Extended Kanban control

Queue $\mathcal{B}_i$ is the output buffer of stage i and contains pairs of stage-i finished parts and stage-i kanbans. Queue $\mathcal{B}_0$ is the raw parts buffer. Queue $\mathcal{D}_i$ contains demands for the production of new stage-i finished parts. Queue $\mathcal{D}$ is the customer demands buffer. Finally queue $\mathcal{K}_i$ contains free stage-i kanbans.

As with the Generalized Kanban control, the Extended Kanban control depends on two parameters per stage, which are stage-i kanbans, k_i, and a basestock level s_i.

In the initial state $\mathcal{B}_i$ contains s_i stage-i finished parts having stage-i kanbans attached to it. Queue $\mathcal{K}_i$ contains k_i - s_i free stage-i kanbans, and all other queues are empty. The Extended Kanban control operates as follows. When a customer demand arrives at the system it is instantaneously split into N + 1 (equals to three in this two stages system) demands: the first demand joins queue $\mathcal{D}$ requesting the release of a finished product from $\mathcal{B}_2$ to the customer, and N (=2) other demands, each one joining the input demand queue $\mathcal{D}_i$ of each stage i, i = 1, 2, requesting the production of a new part in stage i.

When the first demand arrives at $\mathcal{D}$:

- If a part is available in $\mathcal{B}_2$ (which is initially the case), it is released to the customer after detaching the stage-2 kanban. This kanban is then transferred upstream to $\mathcal{K}_2$.
- Otherwise the demand is backordered and has to wait for a finished product to arrive in $\mathcal{B}_2$.

At the same time, when a demand arrives at D_i, i = 1,…, N:

- If there are both part with stage i-1 kanban in $\mathcal{B}_{i-1}$ and stage i kanban in queue $\mathcal{K}_i$, the part is immediately detached from stage i-1 kanban. One stage-i kanban is removed from $\mathcal{K}_i$, attached to the part and the pair is released into $\mathcal{MP}_i$. At the same time the stage i-1 kanban is transferred upstream to $\mathcal{K}_{i-1}$.

- If there is either no part in $\mathcal{B}_{i-1}$ or no stage-i kanban in $\mathcal{K}_i$, the demand is backordered and has to wait in $\mathcal{D}_i$.

There are three differences between the Extended Kanban control and the Kanban control:

First, in an Extended Kanban control, when a customer demand arrives at the system, it is immediately broadcasted to every stage in the system. This implies that each stage in the system knows immediately the need for production of a new part in order to replenish the finished-product buffer.

Second, as in the Generalized Kanban control, in an Extended Kanban control, kanbans move upstream separately from demands. When a stage-i kanban is freed and transferred upstream to queue $\mathcal{K}_i$, it allows the transfer of a finished part from queue $\mathcal{B}_{i-1}$ to $\mathcal{MP}_i$, provided there is a demand for the production of a new stage-i part in $\mathcal{D}_i$. Thus, in the Extended Kanban control the role of the kanbans is only to authorize the transfer of finished parts downstream and not to authorize the transfer of demands upstream, as is the case in the Kanban control.

Third, Extended Kanban control, like the Generalized Kanban control, has two parameters per stage (k_i and s_i). Initially, for each stage i, there are s_i pairs (stage-i finished parts and stage-i kanbans) and k_i - s_i free stage-i kanbans. Again, in the special case where $k_i = s_i$ for all stages, $i = 1,\ldots, N$, the Extended Kanban control and the Kanban control have been shown to be equivalent.

There are three main differences between Extended Kanban control and the Generalized Kanban control:

First, in Extended Kanban control, parts always move together with kanbans, whereas in a Generalized Kanban control, finished parts do not have kanbans attached to them.

Second, in Extended Kanban system, kanbans never participate to the transfer of demands upstream, whereas in a Generalized Kanban system the transfer of demands and the transfer of kanbans are still partially coupled.

Third, in Extended Kanban system, for each stage i, the two parameters k_i and s_i must be such that $k_i > s_i$ in order for the number of free stage-i kanbans initially in queue $\mathcal{K}_i$ to be positive. In Generalized Kanban system, there is no constraint on the two parameters

CHAPTER 3

EXTENDED-CONWIP-KANBAN CONTROL SYSTEM

In this chapter, we describe in detail of a new mechanism for the coordination of multi-stage manufacturing system called Extended CONWIP Kanban control (ECK). This control policy is a combination of the classical Kanban, Base Stock and CONWIP control policies and includes each policy as a special case.

The rest of this chapter is organized as follows. In the next section we describe some motivations from existing controls. Then, we describe the dynamics of the system. In section 3.5 we derive evolution equations describing how the timings of different vents in the ECK system are related to each other, and we describe the effect of the parameters of the system on these timings. Next we present certain important properties of the ECK policy regarding to its production capacity. Finally, conclusions are drawn in section 3.8.

3.1 Motivations from Existing Controls

In Kanban control system, each stage has only one parameter to be specified, the number of kanbans. It plays two opposing roles: (i) to limit the total amount of WIP allowed in each stage, the lesser amount of cards the lower the WIP; and (ii) to provide the target level at the output buffer at each stage to cushion interruptions and variations, the more amount of cards the better. The system can not perform both roles well when demand or effective processing times are highly variable. For instance, in a situation of high demand variability, one would like to have a “large” number of kanbans at times of high demand, to quickly respond to demand. At the same time, one would like to have a “small” number of kanbans at times of low demand, to reduce inventory costs, since the number of kanbans is equal to the target inventory of finished parts. In reality, one would compromise between “large” and “small”. It commonly accepted that kanban control does not work well when demand and the flow of parts are highly variable.

Similar problem occurs under Base Stock control system. One would like to have “large” amount of basestock level to compensate demands during high arrival. On the other hand, since the level of basestock is heavily affect the average WIP level of the system; one would also like to have “small” basestock level at times of low demand. This is because, as long as the WIP limit is not an issue, the basestock level in Base Stock control and the numbers of kanbans in Kanban control play the same role as a target output level at each stage, i.e. the system tries to maintain the same amount of finished parts in the output buffers. However, the dynamics of the two systems are distinctively different.

This problem leads to a need to separate these two roles of WIP limiting and parts buffering into different parameters. As explained in Chapter 2, by passing demand information upstream without requirement of having any part in an output buffer Generalized Kanban control policy partially decouples these two roles by having two parameters per stage which are the number of kanbans and the basestock level. A recently proposed Extended Kanban control policy completely decouples these two roles and also has two parameters per stage by having demand information transmit to all stages directly upon arrival. As a result, the production capacity of the Extended Kanban policy depends only on the number of kanbans at each stage and not on the basestock level of each stage. This is not the case with Generalized Kanban policy where the production capacity of the system depends on both the number of kanbans and the basestock level of each stage. The implication of this property is that designing a good values for the design parameters in Extended Kanban policy is easier than in Generalized Kanban policy.

In this thesis, we designed a hybrid policy called Extended CONWIP Kanban control policy (ECK), which has two parameters per stage, number of kanban and basestock level, and one extra control parameter for the entire system, number of CONWIP. In this system those two roles mentioned are fully decoupled as in the Extended Kanban policy. One could have a “low” basestock of finished parts to keep low inventories in a normal demand context and a fairly “large” number of kanbans and CONWIP to quickly respond to surges in demand. Moreover, in the ECK control system, demands are transmitted to all stages immediately upon their arrival to the system as is the case in the Base Stock and Extended Kanban control systems, so that the demand information has never been blocked.

An important characteristic of the ECK control system is that it introduces CONWIP control parameter for the entire system. There are three reasons behind this introduction of an additional parameter. First, the CONWIP control imposes a stronger WIP control than Kanban control. The main effect of this property is in failure recovery. If a machine fails in a line, the material downstream of it will eventually be removed from the system by the demands. These demands also cause the release of new parts into the system. If the machine stays down long enough, without the CONWIP control, machines upstream of the failure machine will keep producing and accumulate parts in the system until all of the upstream kanban limits is reached which may result in excessive amount of WIP. On the other hand, with CONWIP control mechanism the total number of parts in front of a failure machine is strictly limited to the amount of CONWIP which is generally lower than the total number of kanbans.

Second, imposing CONWIP mechanism results in lower inventory level for the case where processing times at each stage are correlated which may be found in many practical cases. With CONWIP mechanism, the queue lengths at each manufacturing stage are negatively correlated as in a closed queueing network (Whitt 1984), while the queue lengths at each manufacturing stage in Kanban, Base Stock or Extended Kanban controls are uncorrelated. Based on simulation studies in Takahashi and Nakamura (2002), CONWIP control is more efficient under weakly correlated processing times than Kanban control. This leads to a lower average WIP level given the same throughput in CONWIP control system than in Kanban control system.

Lastly, this parameter simplifies the parameter determination process because the production capacity of the system largely depends on only one parameter, we will see more details in Chapter 6. This simplicity can not be achieved in Extended Kanban control, where the production capacity of the system depends on the number of kanbans in all stages, or in Generalized Kanban control system where the production capacity of the system depends on both the number of kanbans and the basestock level of each stage. This property becomes very useful when designing system parameters and adaptive control mechanism for a large system. If the demand change is severe, the intermediate kanbans and basestock levels may need to be adjusted, but the consequences of maladjustments are small. For example, if the numbers of kanbans in all stages are allowed to be infinite and the intermediate basestock levels are zero, the resulting control policy is CONWIP, which performs fairly well.

While installing a new production control can be done easily, implementing it requires fundamental changes in production from existing control to bring it in line with the just-in-time philosophy, and this is far from being easy. The ECK policy we studied can be implemented as a straightforward step-by-step modification to an existing kanban policy. The production manager can decide to either add a CONWIP loop from the last manufacturing stage to the first stage if he concerns more about WIP limiting or implement global demand information flow if he concerns more about demand blockage or separating the two roles of kanban. The first option results in a CONWIP-Kanban control, while the second option results in an Extended Kanban control. Both have been shown to be superior to Kanban, CONWIP or Base Stock control alone. The next step of modification is to implement the last control mechanism which then results in a full ECK control system. In the same way, it can also be implemented as a modification of a CONWIP or Base Stock system.

The ECK control system offers an opportunity to significantly improve the performance of pull production control systems. Some preliminary investigations about the improvement in performance (i.e. a reduction in WIP amount at the same customer service level) of the ECK control system compared with other existing control systems in Chapter 4 indeed confirmed this belief. Robustness is another key issue that also brings out the advantage of the ECK control system. It is a known fact that in a real production environment, the only constant is change. It is therefore very important for any production management policy to be robust to these changes and be able to adapt accordingly. Results from our simulation studies in Chapter 4 and case study in Chapter 8 confirmed this statement.

3.2 System Characteristics

This section describes the card control queues and flow of demand information including contents and initial values of each queue in Extended-CONWIP-Kanban Control System. Figure 3-1 depicts the queueing network model of an ECK system having N stages in series. The contents and initial values of each queue (or network of queues) are shown in Table 3-1.

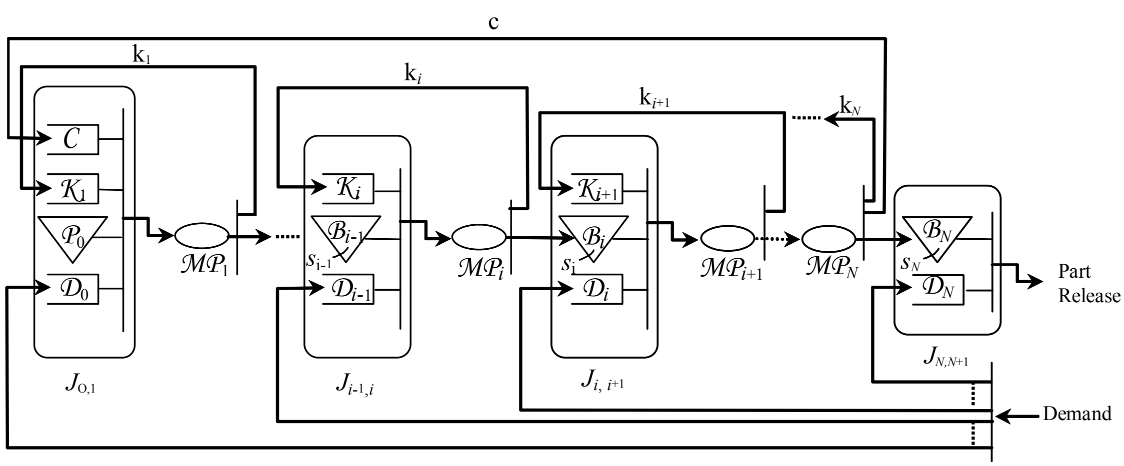


Figure 3-1: Queueing Network Model of an ECK system having N stages in series.

Table 3.1: Contents and initial values of queues in the queueing network model of an ECK system having N stages in series.

Queue		Contents	Initial Value
$\mathcal{MP}_i$	$i = 1,\ldots, N$	(q_i, k_i, c)	0
$\mathcal{B}_i$	$i = 1,\ldots, N$	(p_i, c)	s_i
$\mathcal{K}_i$	$i = 1,\ldots, N$	k_i	k_i
$\mathcal{D}_i$	$i = 1,\ldots, N+1$	d_i	0
$\mathcal{C}$	-	c	$C - \sum_{i=1}^{N-1} s_i$

The ECK control system depends on two parameters per stage and an additional parameter for the entire system. These are stage-i kanbans k_i , a basestock level s_i and the total CONWIP limit, C. Stage- i kanban, k_i, is used to limit the maximum amount of parts in that stage, while stage-i basestock level, s_i, is the target amount of parts at the output buffer of that stage. Initially, s_i amount of stage–i finished parts, p_i, and an equal number of CONWIP card are stored in queue $\mathcal{B}_i$ and k_i amount of stage-i Kanbans are stored in queue $\mathcal{K}_i$. The remaining C - $\sum_i s_i$ CONWIP cards are stored in queue $\mathcal{C}$. Queue $\mathcal{P}_0$ represents the raw material buffer. The initial number of raw material in $\mathcal{P}_0$ and the arrival of new raw material into $\mathcal{P}_0$ fall outside the scope of the control mechanism and are considered as given or infinite.

Production in the system is driven by customer demands. When a customer demand arrives to the system, then the following happens: (i) the demand is splitted into $N + 1$ components, namely d_i, $i = 1,\ldots, N + 1$, and each component is immediately transferred upstream to its respective queue, $\mathcal{D}_i$; (ii) if there is a part in queue $\mathcal{B}_{i-1}$ and a stage–i kanban in queue $\mathcal{K}_i$ (or a raw part in P_0, a stage-1 kanban, and a CONWIP card or just a finished product in queue $\mathcal{B}_N$), the part is labeled q_i and transferred downstream to queue $\mathcal{MP}_i$ (or to the customer if $i = N$), and the demand is satisfied and is therefore dropped. If there is no part in $\mathcal{B}_i$ or no stage–i kanban in $\mathcal{K}_i$ (or no CONWIP card in queue $\mathcal{C}$ if $i = 1$), then the demand, d_i , stays in $\mathcal{D}_i$ as a backordered demand until a part in $\mathcal{B}_i$ and a Kanban in $\mathcal{K}_i$ (and a CONWIP card in $\mathcal{C}$ if $i = 1$) become available.

Queue $\mathcal{B}_i$, $\mathcal{K}_{i+1}$ and $\mathcal{D}_{i+1}$ are joined to form a synchronization station J_i in front of each manufacturing stage. If there is a stage–i finished part, p_i, in queue $\mathcal{B}_i$, a stage–i+1 Kanban, k_{i+1}, in queue $\mathcal{K}_{i+1}$, and a demand, d_{i+1}, in queue $\mathcal{D}_{i+1}$, then: (i) stage– i+1 kanban is attached to part p_i , which is now relabeled q_{i+1} and transferred to queue $\mathcal{MP}_{i+1}$, the part has the CONWIP card attached to it; and (ii) one demand in queue $\mathcal{D}_{i+1}$ is satisfied and is therefore dropped. When a part q_{i+1} finishes its processing at stage i+1, its stage-i+1 kanban k_{i+1} is detached and transferred upstream into queue $\mathcal{K}_{i+1}$, then it is relabeled p_{i+1} and together a CONWIP card move into queue $\mathcal{B}_{i+1}$ except in the last stage where a CONWIP card is detached and move into queue $\mathcal{C}$.

In the last synchronization station $J_{N,\,N+1}$, as soon as there is a finished product in queue $\mathcal{B}_N$ and a demand d_{N+1} in queue $\mathcal{D}_{N+1}$, then: (i) p_N is released to the customer; and (ii) one demand in queue $\mathcal{D}_{N+1}$ is satisfied and is therefore dropped.

In the first synchronization station, as soon as there is a stage-1 Kanban in $\mathcal{K}_1$, a CONWIP card in $\mathcal{C}$, and a demand in $\mathcal{D}_1$, then: (i) the stage–1 Kanban is attached onto raw part; (ii) the CONWIP card is also attached on to the raw part; (iii) the raw part is labeled q_1, and together with the CONWIP card and stage-1 Kanban transferred downstream into $\mathcal{MP}_1$; and (iv) one demand in queue $\mathcal{D}_1$ is satisfied and is therefore dropped.

In conclusion, the dynamics of ECK control system are a combination of the dynamics of the Base Stock control, the CONWIP control, and the Kanban control. In the ECK system, the production is driven by a global demand information flow as is the case in the Base Stock control. The total amount of WIP in the system is limited by the amount of CONWIP as is the case in the CONWIP control. A stage–i finished part, p_i, i = 2,…, N is transferred downstream into $\mathcal{MP}_{i+1}$ (or to the customer if $i = N$) only if one of a finite number of stage–i+1 kanban available in queue $\mathcal{K}_{i+1}$, as is the case in the Kanban control.

3.3 Invariants

In this section, we present the invariants that describe the relationship between the contents of various queues in the ECK control system. Property 3.1 describes the relationship regarding to the kanban control in each manufacturing stage. Property 3.2 describes the relationship regarding to the basestock level of each manufacturing stage. Property 3.3 describes the relationship of queues between two adjacent synchronization stations in term of the number of kanban and the basestock level. Property 3.4 describes the relationship regarding to the CONWIP control. Property 3.5 describes the relationship of queues between the first and the last synchronization stations in term of the number of CONWIP and the summation of basestock levels of each stage. Property 3.6 describes the total amount of part in the system. Property 3.7 describes the relationship between the amount of backorders in the last synchronization station and the amount of orders on-hold in the first synchronization station

Property 3.1: Invariant of kanban control in each stage

$$NQ(\mathcal{K}_i) + NQ(\mathcal{MP}_i) = k_i \qquad i = 1,\ldots, N \tag{3.1}$$

Proof When the ECK system is in its initial state, queue $\mathcal{K}_i$ has k_i kanbans, $\mathcal{MP}_i$ is empty; therefore, Equation (3.1) is true initially. By observing the events that can modify the state of the system, it is clear that as the ECK system evolves, starting from its initial state, Equation (3.1) remains true, since: (i) when the unfinished part q_i leaves $\mathcal{MP}_i$, the stage–i Kanban is detached and moves to queue $\mathcal{K}_i$; (ii) when the finished part, p_{i-1}, joins $\mathcal{MP}_i$, the stage–i kanban moves from queue $\mathcal{K}_i$ to queue $\mathcal{MP}_i$.

The invariant of kanban control in the ECK system is different from that in the Kanban and Extended Kanban. It does not include the amount of parts in the output buffer, $NQ(\mathcal{B}_i)$, in the equation; hence, the stage-i basestock level, s_i, can be independently specified in the ECK system. In the Extended Kanban system, s_i is required to be no greater than k_i. The effect of this relationship constraint will be discussed more in Chapter 4 when we compare the state-space of each policy.

Property 3.2: Invariant of the target basestock at each stage

$$NQ(\mathcal{D}_i) + NQ(\mathcal{MP}_i) + NQ(\mathcal{B}_i) = s_i + NQ(\mathcal{D}_{i+1}) \qquad i = 1,\ldots, N \tag{3.2}$$

Proof When the ECK system is in its initial state, queue $\mathcal{B}_i$ has s_i amount of parts in it while queue $\mathcal{D}_i$, $\mathcal{MP}_i$, and $\mathcal{D}_{i+1}$ are empty; therefore, Equation (3.2) is true initially. By observing the events that can modify the state of the system, it is clear that as the ECK system evolves, starting from its initial state, Equation (3.2) remains true, since: (i) when a part join queue $\mathcal{MP}_i$, a demand in $\mathcal{D}_i$ also leaves; (ii) when a part leaves queue $\mathcal{MP}_i$, it joins queue $\mathcal{B}_i$; (iii) when a part leaves queue $\mathcal{B}_i$, a demand in queue $\mathcal{D}_{i+1}$ also leaves; and (iii) when a demand joins queue $\mathcal{D}_i$, a demand also joins queue $\mathcal{D}_{i+1}$.

This invariant of basestock mechanism is true in all policies that has global demand flow, i.e. Base Stock and Extended Kanban policies

Property 3.3: Relationship of queues between two adjacent synchronization stations in term of the number of kanban and the basestock level

$$[NQ(\mathcal{K}_i) - NQ(\mathcal{D}_i)] - [NQ(\mathcal{B}_i) - NQ(\mathcal{D}_{i+1})] = k_i - s_i \qquad i = 1,\ldots, N \tag{3.3}$$

Proof Equation (3.3) follows after substituting $NQ(\mathcal{MP}_i)$ from (3.1) into (3.2) and rearranging terms.

The relationship here is different from that of the Extended Kanban policy because the $NQ(\mathcal{B}_i)$ term is not included in that case. The reason is that the kanban mechanism in the EK policy linked the two adjacent synchronization stations together and therefore can leave the output parts out of the equation.

Property 3.4: Invariant of the CONWIP control.

$$NQ(\mathcal{C}) + \sum_{i=1}^{N} NQ(\mathcal{MP}_i) + \sum_{i=1}^{N-1} NQ(\mathcal{B}_i) = C \tag{3.4}$$

Proof When the ECK system is in its initial state, queue $\mathcal{C}$ has $C - \sum_{i=1}^{N-1} s_i$ CONWIP cards while all of queues $\mathcal{MP}_i$ are empty and the total numbers of part in $\mathcal{B}_i$, i = 1,…, N - 1 is equal to $\sum_{i=1}^{N-1} s_i$; therefore, Equation (3.4) is true initially. By observing the events that can modify the state of the system, it is clear that as the ECK system evolves, starting from its initial state, Equation (3.4) remains true, since: (i) for every manufacturing stage except the last stage, when a part leaves $\mathcal{MP}_i$, it then joins queue $\mathcal{B}_i$; (ii) for the last manufacturing stage, when a part leaves $\mathcal{MP}_N$, its CONWIP card is detached and returned to queue $\mathcal{C}$; (iii) for every output buffer from i = 1,…, N - 1 when a part leaves $\mathcal{B}_i$, it then joins queue $\mathcal{MP}_{i+1}$; and (iv) for the first synchronization station, a CONWIP card leaves queue $\mathcal{C}$ then is attached to a part which will then join $\mathcal{MP}_1$.

This invariant different from the CONWIP invariants in CONWIP and CONWIP Kanban policy because it does not include the finished goods inventory, $NQ(\mathcal{B}_N)$, in the equation. This is an important feature of ECK policy over the other two policies. It implies that the finished goods inventory and the amount of WIP are fully decoupled. The importance and effects of having this independency will be discussed in Chapter 4.

Property 3.5: Relationship of queues between the first and the last synchronization stations in term of the number of CONWIP and the summation of basestock levels of each stage.

$$[NQ(\mathcal{C}) - NQ(\mathcal{D}_1)] - [NQ(\mathcal{B}_N) - NQ(\mathcal{D}_{N+1})] = C - \sum_{i=1}^{N} s_i \tag{3.5}$$

Proof When the ECK system is in its initial state, queue $\mathcal{C}$ has $C - \sum_{i=1}^{N-1} s_i$ amount of CONWIP cards and queue $\mathcal{B}_N$ has s_N amount of finished parts while queue $\mathcal{D}_1$ and $\mathcal{D}_{N+1}$ are empty; therefore, Equation (3.5) is true initially. By observing the events that can modify the state of the system, it is clear that as the ECK system evolves, starting from its initial state, Equation (3.5) remains true, since: (i) when a demand joins queue $\mathcal{D}_{N+1}$, it also joins queue $\mathcal{D}_1$ because of global information flow; (ii) since queue $\mathcal{B}_N$ and queue $\mathcal{D}_{N+1}$ are in the same synchronization station, when a part leaves queue $\mathcal{B}_N$, a demand also leaves queue $\mathcal{D}_{N+1}$; (iii) since queue $\mathcal{C}$ and queue $\mathcal{D}_{N+1}$ are in the same synchronization station, a CONWIP card will depart from queue $\mathcal{C}$ as soon as a demand leaves queue $\mathcal{D}_1$.

This relationship does not exist in any other control policies. It captures the relationship between the amount of order-on-hold, $NQ(\mathcal{D}_1)$, and the amount of backorders, $NQ(\mathcal{D}_{N+1})$ in term of the control parameters. The benefit of having this relationship is in the demand lead-time quoting because the amount of order-on-hold at the beginning of the line is now more transparent. In the Kanban and Generalized Kanban policy, this amount of order-on-hold is not transparently captured, i.e. the demand information may be delayed somewhere in between the line.

Property 3.6: Total amount of parts in the system

$$\sum_{i=1}^{N} NQ(\mathcal{MP}_i) + \sum_{i=1}^{N} NQ(\mathcal{B}_i) = \sum_{i=1}^{N} s_i - NQ(\mathcal{D}_1) + NQ(\mathcal{D}_{N+1}) \tag{3.6}$$

Proof Equation (3.6) follows after subtracting equation (3.5) from equation (3.4) and rearranging terms.

This relationship is very useful. It implies that we only need to two values which are $NQ(\mathcal{D}_1)$ and $NQ(\mathcal{D}_{N+1})$ for determining the total amount of parts in the system. This is not available in Kanban and Generalized Kanban policies.

Property 3.7 Relationship between the amount of backorders in the last synchronization station and the amount of orders on-hold in the first synchronization station

$$NQ(\mathcal{D}_{N+1}) - NQ(\mathcal{D}_1) = \sum_{i=1}^{N} [k_i - NQ(\mathcal{K}_i)] - \sum_{i=1}^{N} [s_i - NQ(\mathcal{B}_i)] \tag{3.7}$$

Proof Equation (3.7) follows after the summation of equation (3.1) for every stage then substitute the term $\sum_{i=1}^{N} NQ(\mathcal{MP}_i)$ into equation (3.6).

3.4 Bounds

In this section, we present the bounds on the contents of various queues in the ECK control system using invariants in section 3.3. Property 3.8, 3.9 and 3.12 describes bounds resulting from kanban, basestock, and CONIWP mechanisms. Property 3.10 and 3.11 describes bounds at the last and intermediate synchronization stations Property 3.13 describes the relationship between the first and the last synchronization station.

The bounds presented here will be used to construct the state-space transition rate diagram in Chapter 5. They are also important for system designer because they help determine the appropriate control parameters for a given production constraint, e.g. available space at each stage, limit on number of order-on-hold, etc.

Property 3.8: Kanban bound of each manufacturing stage

$$0 \leq NQ(\mathcal{K}_i) \leq k_i \qquad i = 1,\ldots, N \qquad (3.8)$$

$$0 \leq NQ(\mathcal{MP}_i) \leq k_i \qquad i = 1,\ldots, N \qquad (3.9)$$

Proof Inequalities (3.8) and (3.9) follow directly from equation (3.1) since all variables in equation (3.1) are non-negative.

Property 3.9: Basestock bound on the inventory level at each synchronization station

$$NQ(\mathcal{B}_i) - NQ(\mathcal{D}_{i+1}) \leq s_i \qquad i = 1,\ldots, N \qquad (3.10)$$

Proof Inequality (3.10) follows directly from equation (3.2) since all variables in equation (3.2) are non-negative.

Property 3.10: Basestock bound at the last synchronization station

$$0 \leq NQ(\mathcal{B}_N) \leq s_N \tag{3.11}$$

Proof Since the last synchronization station has only two queues either queue $\mathcal{B}_N$ or queue $\mathcal{D}_{N+1}$ has to be zero when the other queue is positive. Inequality (3.11) follows from inequality (3.10) evaluated at $i = N$.

Property 3.11: Bound on the inventory position at each synchronization station

$$s_i - k_i - NQ(\mathcal{D}_i) \leq NQ(\mathcal{B}_i) - NQ(\mathcal{D}_{i+1}) \leq s_i - NQ(\mathcal{D}_i) \quad i = 1,\ldots, N \tag{3.12}$$

Proof Inequality (3.12) follows after substituting $NQ(\mathcal{K}_i)$ from equation (3.3) into inequality (3.8) and rearranging terms.

Property 3.12: CONWIP bounds

$$0 \leq NQ(\mathcal{C}) \leq C \tag{3.13}$$

$$0 \leq \sum_{i=1}^{N} NQ(\mathcal{MP}_i) \leq C \tag{3.14}$$

$$0 \le \sum_{i=1}^{N-1} NQ(\mathcal{B}_i) \le C \tag{3.15}$$

$$0 \le \sum_{i=1}^{N} NQ(\mathcal{MP}_i) + \sum_{i=1}^{N-1} NQ(\mathcal{B}_i) \le C \tag{3.16}$$

Proof Inequalities (3.14), (3.15), and (3.16) follow directly from equation (3.4) since all variables in equation (3.4) are non-negative.

Property 3.13: Relationship between finished goods inventory position and the number of order on-hold in the first stage in term of the number of CONWIP and the total amount of basestock

$$\sum_{i=1}^{N} s_i - C - NQ(\mathcal{D}_1) \le NQ(\mathcal{B}_N) - NQ(\mathcal{D}_{N+1}) \le \sum_{i=1}^{N} s_i - NQ(\mathcal{D}_1) \tag{3.17}$$

Proof Inequality (3.17) follows after substituting $\sum_{i=1}^{N} NQ(\mathcal{MP}_i) + \sum_{i=1}^{N-1} NQ(\mathcal{B}_i)$ from equation (3.6) into inequality (3.16) and rearranging terms.

3.5 Boundary cases

In this section, we presents boundary cases of the ECK control where the ECK control is reduced to the Base Stock control, the CONWIP control, and the Kanban control. From these, we can understand how to modify the basic controls mention above and upgrade them into the ECK control.

Property 3.14: The ECK control system with $k_i = \infty$, $i = 1,\ldots, N$, $c = \infty$, and $s_i \geq 0$, $i = 1,\ldots, N$, is equivalent to the Base Stock control having s_i target output buffer in stage i.

Proof Consider the ECK control system shown in Figure 3-1, with $k_i = \infty$, $i = 1,\ldots, N$, $c = \infty$, and $s_i \geq 0$, $i = 1,\ldots, N$. Queue $\mathcal{K}_i$ and $\mathcal{C}$ have an infinite number of Kanban and CONWIP cards therefore play no role in the synchronization station they belong to, since they never block the passage of parts through that synchronization stations; hence they can be eliminated. Once queues $\mathcal{K}_i$ and $\mathcal{C}$ are eliminated from the queuing network, the remaining network is the same as the queuing network of the Base Stock control system and has the same initial conditions.

Property 3.15 The ECK control system with $k_i = \infty$ (or at least $k_i = C$, $i = 1,\ldots, N$), $s_i = 0$, $i = 1,\ldots, N-1$, $s_N = C$ and $C \geq 0$ is equivalent to the CONWIP control system having C CONWIP cards in the system.

Proof Consider the ECK system shown in Figure 3-1, with $k_i = \infty$, $i = 1,\ldots, N$, $s_i = 0$, $i = 1,\ldots, N-1$, $s_N = C$ and $C \geq 0$. Queue $\mathcal{K}_i$ have an infinite number of kanbans and therefore play no role in the synchronization station they belong to, since they never block the passage of parts through that synchronization stations; hence they can be eliminated. For the remaining queuing network, in the initial state of the system there are no parts in the output buffers, $\mathcal{B}_i$, $i = 1,\ldots, N$, except the final stage $\mathcal{B}_N$ having $s_N = C$ finished parts. Since $s_i = 0$, $i = 1,\ldots, N - 1$, Equation (3.3) implies that $NQ(\mathcal{D}_{i+1}) \geq NQ(\mathcal{B}_i)$, $i = 1,\ldots, N$. Therefore, queues $\mathcal{D}_i$, $i = 2,\ldots, N$ play no role in the synchronization station they belong to; hence they can be eliminated. Once queues $\mathcal{D}_i$, $i = 2,\ldots, N$ are eliminated, an order is released into the production line only if there is at least one finished part in queue $\mathcal{B}_N$ at the time of a demand arrival. This means that an order is released into the production line only when a part in the finished goods queue is consumed. The resulting policy is equivalent to the CONWIP control system having C CONWIP cards in the system.

Property 3.16: The ECK control system with $C = \infty$ (or at least greater than or equal to the sum of all Kanbans in the system) and $k_i = s_i$, $i = 1,\ldots, N$ is equivalent to the simple Kanban control system having k_i kanbans in stage i.

Proof Consider the ECK control system shown in Figure 1, with $c = \infty$ and $k_i = s_i$, $i = 1,\ldots, N$. Queue $\mathcal{C}$ has an infinite number of CONWIP cards, and therefore play no role in the synchronization station it belongs to, since it will never block the passage of parts through that synchronization station; hence it can be eliminated. Once queues $\mathcal{C}$ is eliminated, at each manufacturing stage an order is released into that stage only if there is at least one part in its output queue at the time of a demand arrival. This means that an order is released into each manufacturing stage only when a part in the output queue is consumed. The resulting policy is equivalent to the Kanban control system having k_i stage – i Kanbans.

3.6 Variation of the Extended CONWIP Kanban Control

In this section, we provide three variations of the ECK control system, each representing a different approach on WIP control. There are three queues in each stage where WIP control can be involved. They are input buffer (IP), manufacturing process (MP), and output buffer (OP). Input buffer of stage i is the buffer $\mathcal{B}_{i-1}$, manufacturing process of stage i is $\mathcal{MP}_i$, and output buffer of stage i is the buffer $\mathcal{B}_i$.

The general form of ECK control system presented earlier in this chapter can be considered as input buffer and manufacturing process (IP-MP) WIP control because the kanban mechanism only controls the amount of parts in $\mathcal{B}_{i-1}$ and $\mathcal{MP}_i$ and does not include the output buffer, $\mathcal{B}_i$, in the control. The CONWIP mechanism is also an IP-MP WIP control since it does not include the finished goods inventory in the control.

The first variation of ECK control system is IP-MP-OP WIP control as shown in Figure 3-2. Here the kanban mechanism in each stage controls over $\mathcal{B}_{i-1}$, $\mathcal{MP}_i$ and $\mathcal{B}_i$. The CONWIP mechanism here also includes the finished goods inventory, $\mathcal{B}_N$ in it. This can be considered as a modification of Extended Kanban system by imposing the CONWIP limit over the entire production line. The shortfall of this variation is that the basestock level at each stage is now constrained to be no greater than the amount of kanbans in that stage, which causes less flexibility in the parameter configuration. The advantage of this variation is that kanbans cards are attached to parts at all time which may help reducing confusion in the practical situation.

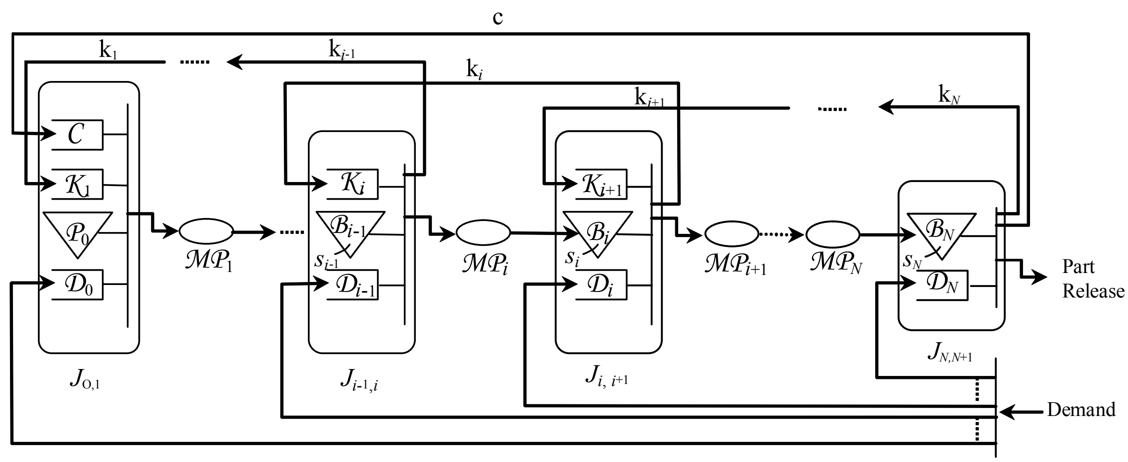


Figure 3-2: ECK control system with IP-MP-OP WIP control

Figure 3-3 shows the second variation of ECK system. It has only manufacturing process (MP) WIP control. Here the kanbans limit only the amount of parts being processed. The advantage of this system is that the authorization of parts by demand information and the release of part based on kanban constraint are distinctively separated. The disadvantage of this system is that it requires double amount of synchronization station which can cause confusion in reality.

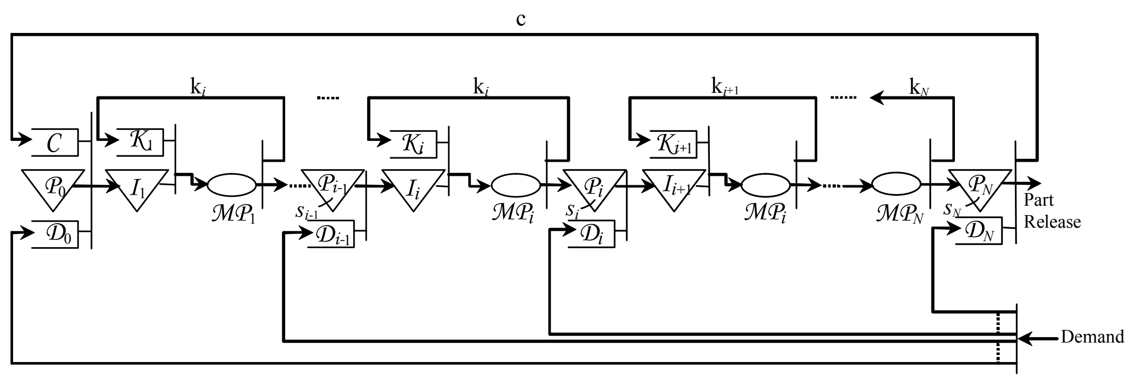


Figure 3-3: ECK control system with MP WIP control

The last variation of ECK system is the one having MP-OP WIP control as shown in Figure 3-4. The kanban mechanism is now limiting the amount of part being processed and the amount of part in the output buffer. The advantage of this variation is the same as in the MP WIP control case in that the authorizing and releasing of parts are separated. However, the disadvantage of this variation is that the basestock level is constrained to be no greater than the kanban levels similar to the IP-MP-OP case.

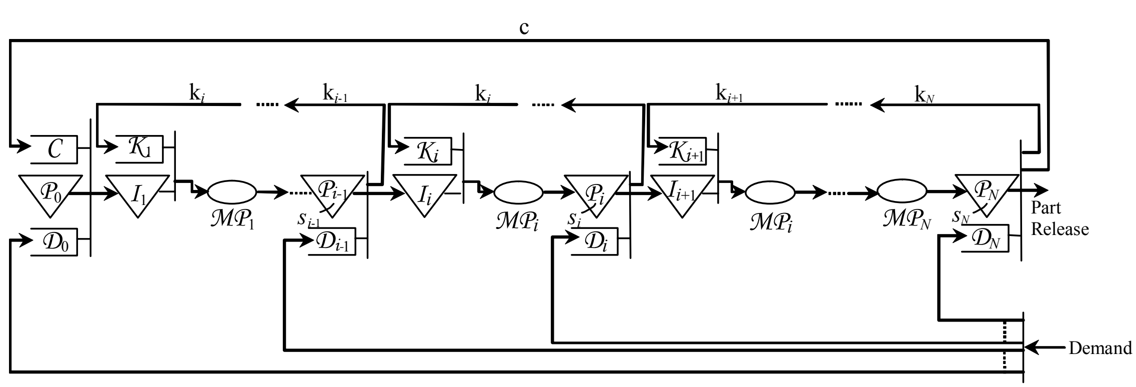


Figure 3-4: ECK control system with MP-OP WIP control

CHAPTER 4

SIMULATION COMPARISIONS

In this chapter, we provide the performance comparisons among pull-type production control policies discussed in Chapter 2 and 3. The purposed of this chapter is to compare the performance of the new class of control, the ECK policy, against the performances of existing policies under various production situations. The results of this study will help choosing the policy to be implemented for a given production scenario.

This chapter is organized as follow. The model assumption and performance measures will be presented in the following section. Details of our simulation model are then presented. In section 4.4, we provide the comparison results from three selective cases. Finally, a discussion and conclusion on the behavior of each control policy and suggestion on implementing strategy is presented in the last section.

4.1 Model assumptions

Arena simulation software package version 7.0 was used to do the simulation experiments in this thesis. The control policies we compared are Simple Kanban (SK), Basestock (BS), CONWIP (CW), CONWIP Kanban (CK), Generalized Kanban (GK), Extended Kanban (EK), and Extended CONWIP Kanban (ECK).

We assume the following assumptions to focus on essential aspects of comparison and to limit the scope of the study:

- The system makes a single part type.
- There is no setup times at each machine
- Material is transported in units of one with ignorable transfer time.
- Information flows instantaneously.
- Parts authorized for loading follow a first come first serve (FIFO) dispatching policy at all machines.
- The time to failure of each machine is operation dependent failures. This implies that no machine can fail unless it is working on a part.
- Any demand that cannot be satisfied immediately from finished goods inventory is backlogged.

4.2 Performance Measures

An important performance measure in the pull production system is the *service level* (or fill rate). This is the fraction of all demands that find a finished product ready for use upon their arrivals. Another important performance measure is the amount of *total inventory* in the system. We define the total inventory as the total amount of parts that have been released to the first machine, but have not yet leaved the system to satisfy demand. We do not consider parts that are authorized for loading at the first stage until they are actually released.

This is a multi-attribute decision making problem having two objectives which are maximizing service level and minimizing the inventory. Specifically, for each control policy, we find the parameter combination that minimize the total inventory for every possible service level, α, or

$$\min_{\{s_i, k_i, C\}} \mathrm{WIP}(s_i, k_i, C)$$

$$\text{s.t.} \quad \text{Service level} \geq \alpha$$

We then plot the performance from the best parameter combination for each policy. This is the tradeoff curves or efficient frontiers which are the convex hulls of all service level/inventory pairs obtained from each control policy. Hence, each point on these curves corresponds to the parameter choice within a particular policy that achieved that service level with the least amount of inventory. All the other parameter combinations gave data points inside the convex hull, and therefore gave less efficient usage of the inventory.

4.3 Simulation models

We studied a two-stage production system having two machines per stage, similar to the queueing network models shown in chapter 2. We construct three cases. In all cases, the processing time is deterministic at one part per unit time. The mean time between demand arrivals is 1.5 time unit. This yields the demand rate of 0.67 parts per unit time. The meantime to failure and mean time to repair are defined such that the machine availability is 90% for all cases. We fix all these parameters to investigate the performances only when the down time is prolonged or the arrival process is stochastic.

The three cases are (*i*) base case, (*ii*) case with "lumped" repair times, and (*iii*) case with Poisson arrival process, see Table 4.1 for specific parameters. The base case is where the mean time to failure (MTTF) and mean time to repair (MTTR) of each machine are exponentially distributed with mean 9 and 1 time unit and the demand process is deterministic. In case 2, we lumped the MTTF and MTTR of each machine proportionally to be 27 and 3 time units. Although the lumping does not change the machine availability, it increase the coefficient of variation of effective processing time (Hopp and Spearman, 1996). The demand process of case 2 remains deterministic. In case 3, we use failure parameter as in the base case but increase the demand arrival variation by changing the demand process to be exponentially distributed with mean inter-arrival time of 1.5 time unit.

Table 4.1: Three performance comparison cases

		Processing Time Variation		Demand Distribution
		MTTF	MTTR	
Case 1: Base Case		Expo (9)	Expo (1)	Deterministic
Case 2: Increase Processing Time Variation		Expo (27)	Expo (3)	Deterministic
Case 3: Increase Demand Variation		Expo (9)	Expo (1)	Exponential

Each simulation was run for 5 replications with 110,000 simulation time unit and initial warm-up period of 10,000 time unit. This gave half widths of the 95% confidence intervals on the estimated values of service level and total inventory of less than 1% in case 1, 2% in case 2, and 1.5% in case 3 from their respective estimated values.

4.4 Simulation results

Case 1: Deterministic demand and deterministic service time with random failure

This is the base case of our simulation study. The configuration of control parameters for each control policy is shown in Table 4.2. For CONWIP policy, we simulated 9 parameter sets, $4 \leq C \leq 12$. This gave the range of service level from 58% to 99.98%. This also provided a starting point on the scope of the parameters of other control policy.

For Base Stock policy, we simulated the parameter sets of $\{s_1, s_2 \mid 0 \leq s_1 \leq 2, 4 \leq s_2 \leq 12\}$. We pick these ranges from experience and trial and error. We found that having $s_1 \geq 3$ results in too much excessive inventory. The reasoning for choosing a larger last buffer is that inventory is not available to satisfy demand unless it is in the last buffer, so we want to keep high level of WIP there. Since the holding costs in different stages are assumed to be the same, holding parts in the last buffer will give better impact to the service level of the system.

For Kanban policy, we simulated the parameter sets of $\{k_1, k_2 \mid 2 \leq k_1 \leq 4, 3 \leq k_2 \leq 10\}$. We found that having the amount of kanbans in stage 1 less than 2 could not satisfy the demand rate and having the amount of kanbans in stage 1 greater than 4 results in too excessive inventory. Since the maximum amount of CONWIPs in the CONWIP policy is 12, the maximum amount of kanbans in stage2 should be less than or equal to 12. We found that having the limit on stage-2 kanbans to be greater than 10 results in service level greater than 99.98% for any amount of sage -1 kanbans; hence, we limit the amount of stage-2 kanbans to be 10.

For the hybrid policies, we used the ranges of parameters found in the traditional policies as a guideline for the range of parameters to be used in the hybrid policies. For CONWIP Kanban policy, in order to utilize both the CONWIP and the Kanban mechanisms, the sum of kanbans in each stage should not be less than the amount of CONWIPs. In addition, having the amount of kanbans in each stage greater than the amount of CONWIPs results in the same result as having the amount of kanbans in that stage equal to the amount of CONWIPs. Thus, we eliminated the cases where the amount of kanbans exceeds the amount of CONWIPs.

The number of cases simulated for Extended Kanban policy was lower than those in Generalized Kanban policy because the kanban mechanism in Extended Kanban required having the basestock level less than the amount of kanbans in each stage. On the other hand, the basestock levels and the amount of kanbans in each stage are independent in Generalized Kanban policy. For the Extended CONWIP Kanban policy, we first simulated the cases where the basestock levels at stage 2 are even numbers to determine configurations that can be part of the tradeoff curve. We then simulated all possible combination of parameters for those candidate configurations.

Table 4.2: Parameter configuration for case 1

Case 1	CONWIP	Kanban 1	Kanban 2	Basestock 1	Basestock 2	# of cases
CONWIP	4 - 12	-	-	-	-	9
Base Stock	-	-	-	0 - 2	4 - 12	27
Kanban	-	2 - 4	3 - 10	-	-	24
CONWIP Kanban	4 - 10	2 - 4	3 - 10	-	-	105
Extended Kanban	-	2 - 4	3 - 10	0 - 2	3 - 10	324
Generalized Kanban	-	2 - 4	3 - 10	0 - 2	3 - 10	576
Extended CONWIP Kanban	4 - 10	2 - 4	2 - 9	0 - 2	3 - 10	2097*

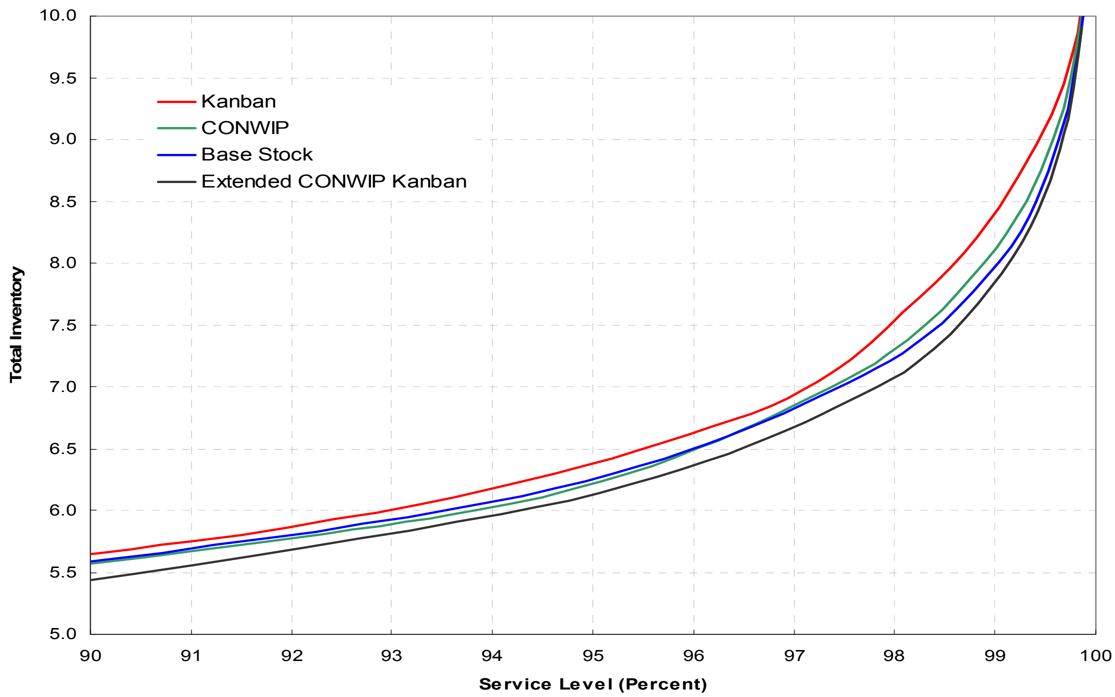
* partial enumeration

Figure 4-1 (a) shows the tradeoff curves comparing traditional policies which are Kanban, CONWIP and Base Stock with the Extended CONWIP Kanban policy. Figure 4-1 (b) shows the tradeoff curves comparing hybrid policies with the Extended CONWIP Kanban policy. A dashed line representing the tradeoff curve from Kanban policy is also presented in Figure 4-1 (b) to provide relative comparison with traditional policies. In this case, the policy provides best performance is the Extended CONIWP Kanban policy and the worst is the Kanban policy.

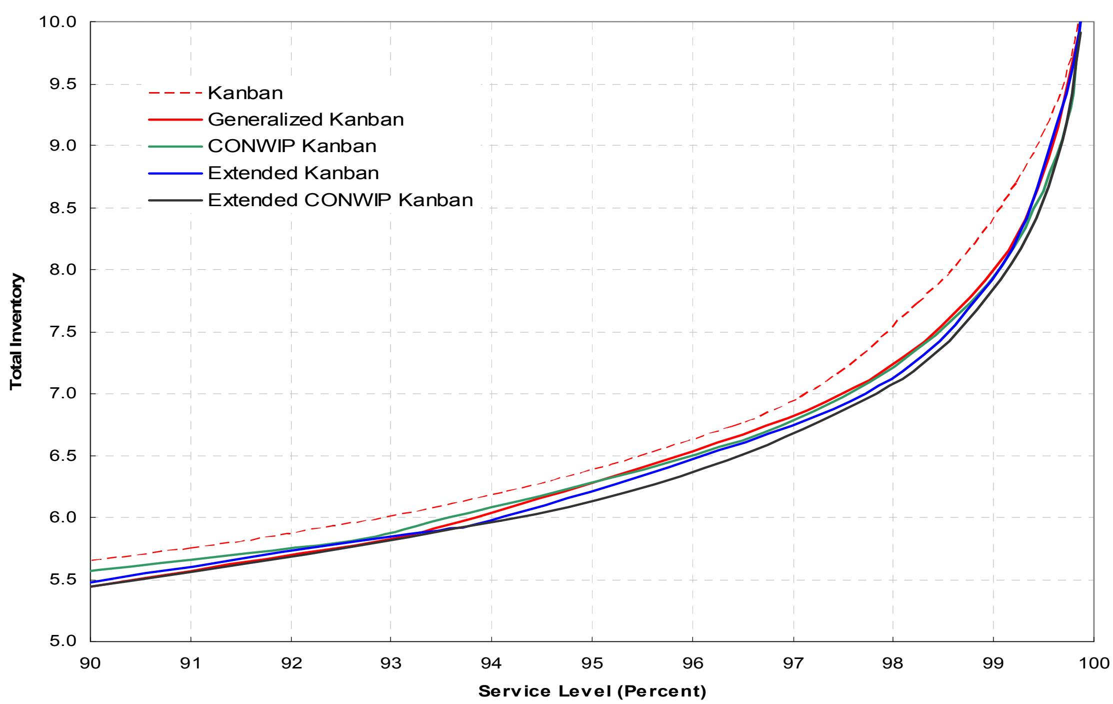
The vertical distance between the curves is the difference in inventory required to attain a particular service level. This difference becomes larger as the required service level increases up to about 99%. Once the required service level approaches 100%, the difference in performances of each control policy becomes insignificant. The average difference between the best and the worst policies is about 5%.

From Figure 4-1 (a), both CONWIP and Base Stock policies provide better service level/inventory compromise than Kanban at all service levels. This result is very important: It shows that a simpler control such as CONWIP control can perform better than a well-known Kanban control through the changes in the flow of demand information to release the raw materials.

The Base Stock policy performed slightly better at higher service level and slightly worse at lower service level than the CONWIP policy. Note that since we did not consider the variation in the WIP level as one of the performance measure here; hence, the lack of WIP control in Base Stock policy does not deteriorate its performance in this comparison.



(a)



(b)

Figure 4-1: Tradeoff between service level and inventory (Case 1)

From Figure 4-1 (b), the performances of all hybrid policies fall between the Kanban and the ECK policies. The performance of Extended Kanban policy seems to be slightly better than CONWIP Kanban and Generalized Kanban in the majority of service level range. However, none of the policies could achieve lower inventories at any service level than the Extended CONWIP Kanban policy.

Case 2: Lumped failure process

In this case the variation of processing time increased by changing the MTTF and MTTR from Expo (9) and Expo (1) to Expo (27) and Expo (3) respectively. The configuration of control parameters are given in Table 4.3. In this case, the range of the control parameters is higher and wider then those in the base case because the higher processing variation required higher amount of WIP in the system.

Table 4.3: Parameter configuration for case 2

Case 2	CONWIP	Kanban 1	Kanban 2	Basestock 1	Basestock 2	# of cases
CONWIP	8 - 24	-	-	-	-	17
Base Stock	-	-	-	0 - 3	6 - 24	76
Kanban	-	2 - 5	8 - 24	-	-	76
CONWIP Kanban	8 - 24	2 - 5	8 - 24	-	-	748
Extended Kanban	-	2 - 5	8 - 24	0 - 2	6 - 24	1404*
Generalized Kanban	-	2 - 5	8 - 24	0 - 2	6 - 24	1860*
Extended CONWIP Kanban	8 - 20	2 - 4	6 - 18	0 - 2	6 - 24	3504*

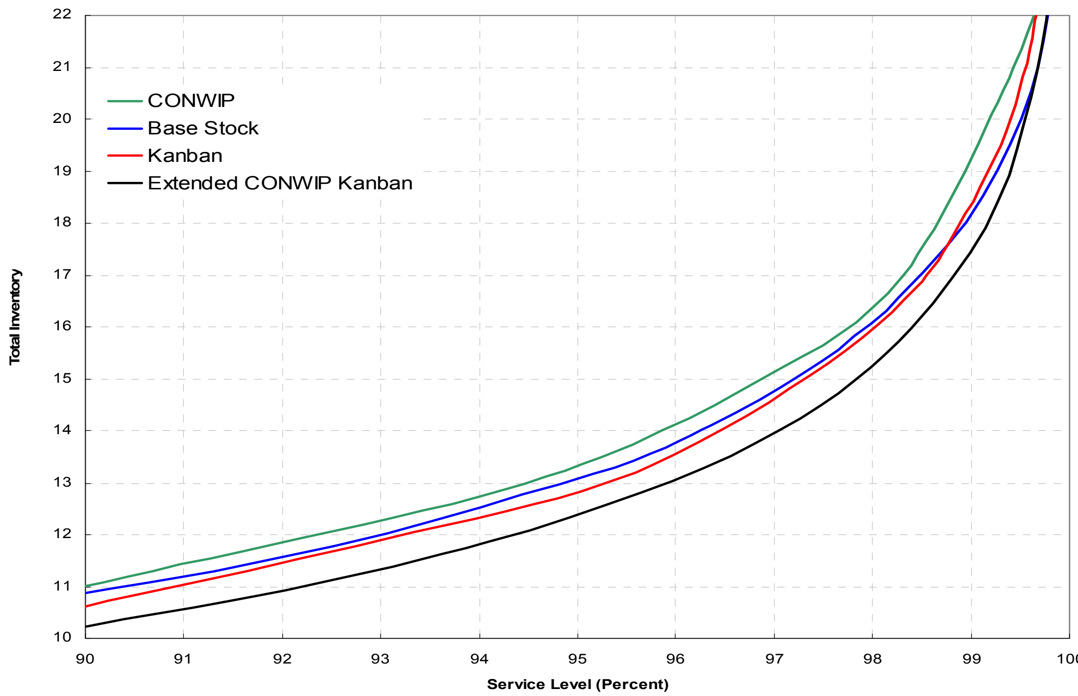
* partial enumeration

Again, we started with CONWIP policy which varied its CONWIPs from 8 to 24. We then enumerated and simulated the Base Stock and Kanban cases. We used the parameter configurations found in these traditional control policies to determine the range of parameters in the hybrid policies.

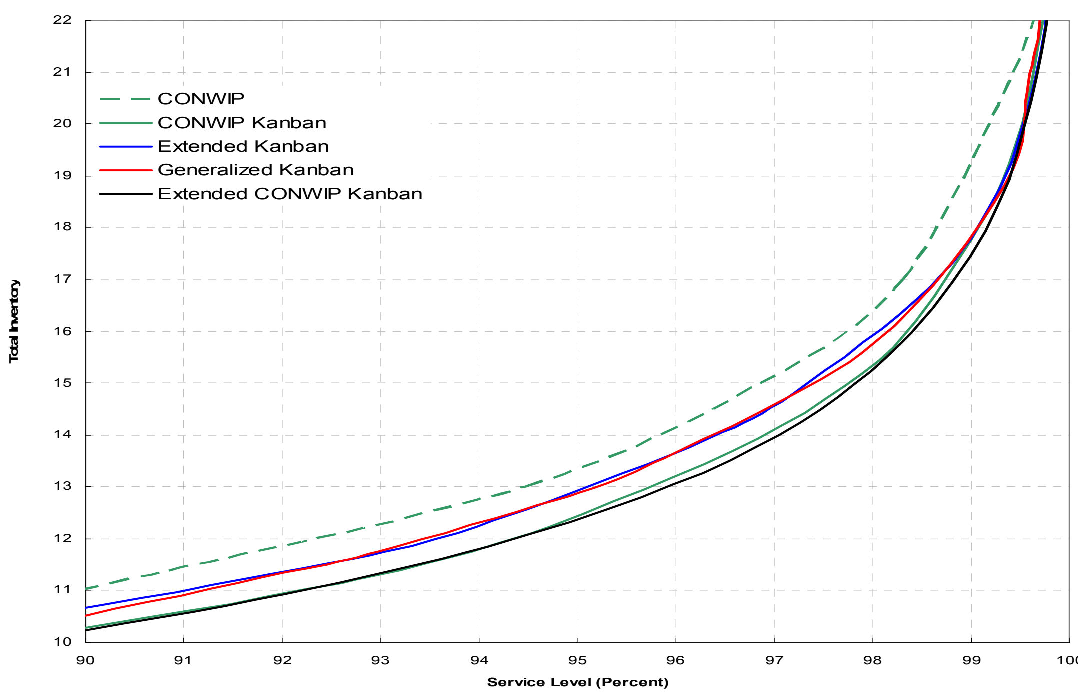
Due to enormous amount of possible configurations, we decided to simulate partial enumeration for the hybrid policies. For the Extended Kanban and Generalized Kanban policies, we enumerated and simulated the cases where the basestock at stage 2 was an even number first. Using the results found, we determined good candidate configurations and simulated all other missing configurations. Similarly, for ECK policy, we enumerated and simulated the cases where the amount of stage-2 kanbans and the stage-2 basestock are even numbers first, then simulated other missing candidate configurations.

Figure 4-2 (a) shows the tradeoff curves comparing traditional policies with the ECK policy. Figure 4-2 (b) shows the tradeoff curves comparing hybrid policies with the ECK policy. Here, a dashed line representing the tradeoff curve from CONWIP policy is also presented to provide a lower bound and relative comparison with traditional policies.

In this case, the best performing policy is again the Extended CONWIP Kanban policy and the worst is now the CONWIP policy. The average difference between the best and the worst policies is about 8% which is greater than the one in the base case.



(a)



(b)

Figure 4-2: Tradeoff between service level and inventory (Case 2)

From Figure 4-2 (a), Kanban and Base Stock policies make better service level/inventory compromise than CONWIP policy at all service levels. Unlike the results in the base case, the result in this case shows that the correlated WIP control in CONWIP policy can not perform well in a high process variation situation. Dedicate control points either through Kanban or basestock mechanisms results in better performance. This will be discussed in detail in Chapter 5 when we analyze the effect of adjusting the state space of the system. The Base Stock policy performed slightly better at very high service level and slightly worse at lower service level than the Kanban policy. The reasoning is that the WIP control in Kanban policy is more important at lower service level in order to achieve lower inventory; however, the possibility of demand blockage decrease its performance in the very high service level region.

From Figure 4-2 (b), the performances of all hybrid policies fall between the Kanban and the ECK policies. The performance of CONWIP Kanban policy was very close to the performance of the best policy, the Extended CONWIP Kanban policy. On the other hand, the performance of Extended Kanban and Generalized Kanban are almost identical and in the middle between the best and worst policy. This is an interesting result. Even though CONWIP policy is the worst policy in this case, by combining CONWIP mechanism with other mechanisms yielded the best result. The reason is as follow. Since the manufacturing process has more variability, it is required to have large amount of kanbans or WIP limit in each stage in order to compensate the process variability. However, if there is no correlation control between the amounts of WIP in each stage through CONWIP mechanism, the release of new raw material might be excessive and the new part will be released into the system even though there are plenty amount of part downstream already.

Case 3: Increase Demand Variation

In this case the demand variation increased by having the demand inter-arrival time to be exponentially distributed instead of deterministic as in the base case. The configuration of control parameters are given in Table 4.4. In this case the range of control parameter is slightly lower than the range in case 2.

Table 4.4: Parameter configuration for case 3

Case 3	CONWIP	Kanban 1	Kanban 2	Basestock 1	Basestock 2	# of cases
CONWIP	8 - 20	-	-	-	-	13
Base Stock	-	-	-	0 - 2	8 - 20	39
Kanban	-	2 - 4	8- 20	-	-	39
CONWIP Kanban	8 - 20	2 - 4	8- 20	-	-	273
Extended Kanban	-	2 - 4	8- 20	0 - 2	6 - 20	936
Generalized Kanban	-	2 - 4	8- 20	0 - 2	6 - 20	971*
Extended CONWIP Kanban	8 - 20	2 - 4	4 - 18	0 - 1	8 - 20	2340*

* partial enumeration

The amount of CONWIPs in the CONWIP policy for this case varied from 8 to 20. In the Base Stock system, the basestock level at stage 1 varied from 0 to 2 and the basestock at stage 2 varied from 8 to 20. For the Kanban policy, the amount of stage-1 kanbans varied from 2 to 4 while stage-2 kanban varied from 8 to 20.

For the CONWIP Kanban and Extend Kanban policies, we did total enumeration and simulated all possible configurations. We did simulate partial enumeration for the Generalized Kanban and Extended CONWIP Kanban policies. For the Generalized

Kanban policy, we enumerated and simulated the cases where the basestock at stage 2 was an even number first. Then, we determined good candidate configurations and simulated all other missing configurations. For ECK policy, we enumerated and simulated the cases where the amount of stage-2 kanbans and the stage-2 basestock are even numbers first, then simulated other missing candidate configurations.

Similar to case 2, the best performing policy is the Extended CONWIP Kanban policy and the worst is the CONWIP policy. Figure 4-3 (a) shows the tradeoff curves comparing traditional policies with the Extended CONWIP Kanban policy. Figure 4-3 (b) shows the tradeoff curves comparing hybrid policies with the Extended CONWIP Kanban policy. Here, a dashed line representing the tradeoff curve from CONWIP policy is also presented to provide a lower bound and relative comparison with traditional policies.

By looking closer at Figure 4-3 (a), a different result from case 2 arise. The CONWIP policy is still the worst and the Kanban policy is still the best among the three tradition policies; however, the performance of the Base Stock policy is now closer to the CONWIP policy which is different from the result in case 2. This indicates that passing the demand information directly to release the raw material is worse than having them passing upstream stage by kanban mechanism. The reasoning is that the demand process has more variability than the manufacturing process. Thus, direct demand flow will pass the high variability from demand process to the raw material releasing process.

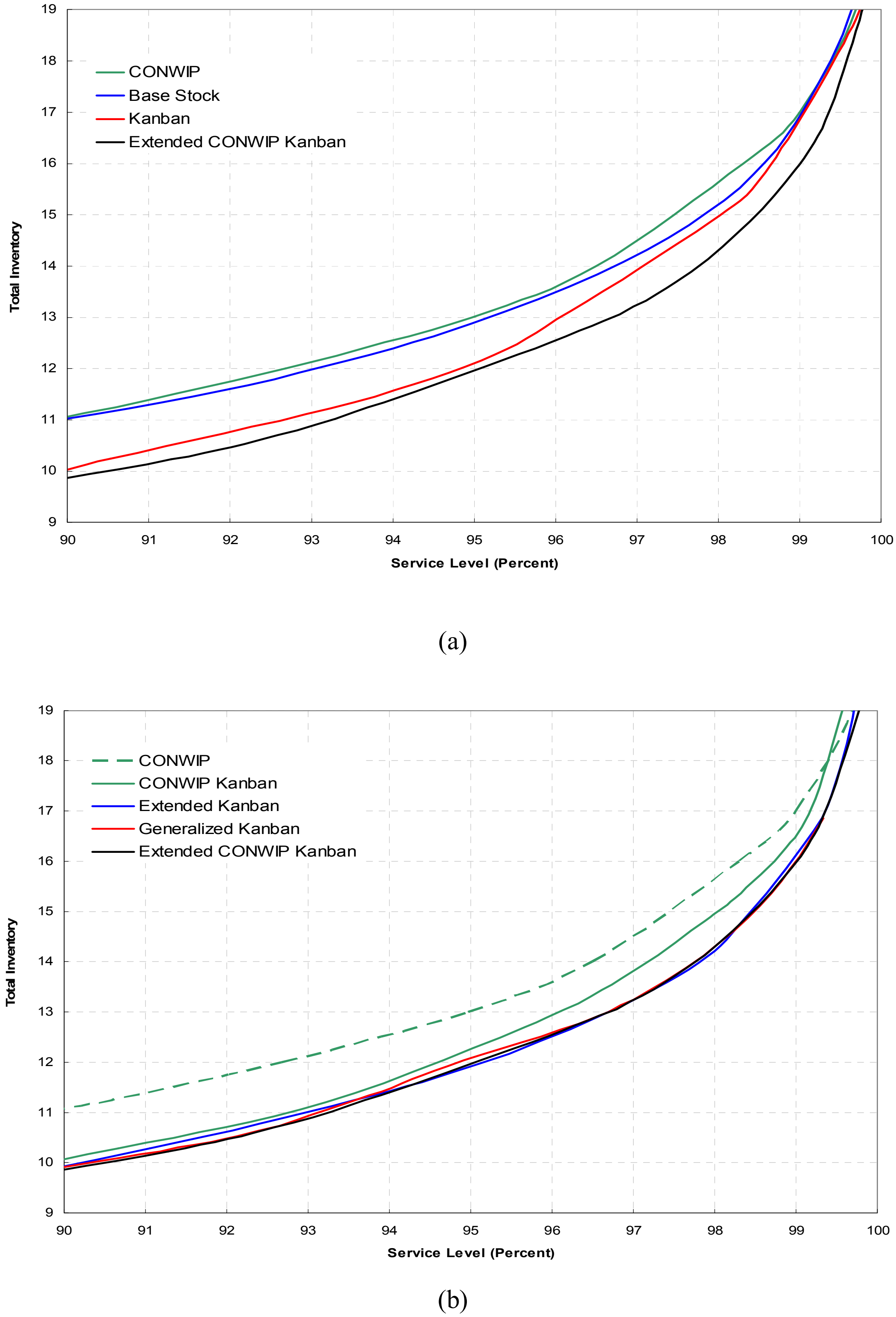


Figure 4-3: Tradeoff between service level and inventory (Case 3)

From Figure 4-3 (b), the performances of all hybrid policies fall between the Kanban and the ECK policies. Opposite result from case 2 appeared here. The performances of Extended Kanban and Generalized Kanban policies were very close to the performance of the best policy, the Extended CONWIP Kanban policy. Instead, the performance of the CONWIP Kanban is now the worst among hybrid policies. The reason is that the CONWIP mechanism in the CONWIP Kanban policy resulted in higher variation in raw material releasing than other hybrid policies.

4.5 Discussion

It is interesting to see that even when the process time, process availability, arrival rate are kept the same, the performance of each control policy can vary significantly. The reasons for the performance difference between these control policies are mainly the results of the difference in the demand information flow and the WIP control mechanism. Table 4.5 summarizes and compares these differences.

Demand information flow can be categorized as coupled, partially coupled or uncoupled with the WIP control mechanism. In Kanban control, information flows upstream only when material is moved downstream which is considered as fully coupled with the WIP mechanism. This is somewhat relaxed in the CONWIP, CONWIP Kanban, and Generalized Kanban policies. In CONWIP and CONWIP Kanban policies, the demand information is sent to the first machine via the CONWIP cards, but only when the finished goods inventory is non-empty. In Generalized Kanban policy, the transfer of demand upstream is not completely synchronized with the transfer of parts downstream (see Chapter 2 for more details). The other policies distribute demand information directly to all machines whenever demands occur.

For the WIP control mechanism, this can be categorized as local or global WIP control. Local WIP control limits the maximum amount of WIP in each stage individually, while global WIP control limits the maximum amount of WIP. Therefore, the amounts of WIP in the production line in each location are more correlated if the global WIP control is implemented.

Table 4.5: Classification of control policies

Control Policies	Demand Information Flow			WIP Control	
	Coupled	Partially Coupled	Uncoupled	Local	Global
Traditional Policies					
CONWIP		✓			✓
Base Stock			✓		
Kanban	✓			✓	
Hybrid Policies					
CONWIP Kanban		✓		✓	✓
Extended Kanban			✓	✓	
Generalized Kanban		✓		✓	
Extended CONWIP Kanban			✓	✓	✓

In Table 4.6, we evaluated the performances of each control policy in each simulation cases and categorized them in to three levels: good, better, and best. The purpose of this evaluation is to compare and link these results with the classification of demand flow and WIP control presented in Table 4.5. It should be noted that we separated the comparison between traditional and hybrid policies. Since all hybrid policies contain kanban control, the effect of local WIP control is inherited in every hybrid controls which cause them to be compared separately from traditional ones.

Table 4.6: Comparison of control policies

Control Policies	Case 1 Base case	Case 2 Higher process variation	Case 3 Higher demand variation
Traditional Policies			
CONWIP	✓✓	✓	✓
Base Stock	✓✓	✓✓	✓
Kanban	✓	✓✓	✓✓
Hybrid Policies			
CONWIP Kanban	✓✓	✓✓✓	✓✓
Extended Kanban	✓✓✓	✓✓	✓✓✓
Generalized Kanban	✓✓	✓✓	✓✓✓
Extended CONWIP Kanban	✓✓✓	✓✓✓	✓✓✓

✓ = good, ✓✓ = better, ✓✓✓ = best

The advantage of decoupling the information flow from the parts movements is that it gives added stability against disruptions along the line. This is the reason why CONWIP and Base Stock outperform Kanban in case 1. This effect becomes more important as the utilization of the system increases. However, when the demand variation is high, the passage of demand flow without any mechanism to limit the release of raw material cause higher part releasing variability and lower performance. This can be seen in the comparison among traditional policies in case 3. For the hybrid policies, the uncoupled demand flow in EK and ECK led to better performance than the partially coupled demand flow in CK and GK in case 1.

The advantage of having local WIP control is shown off in case 2 where manufacturing process has higher variation. This causes the Kanban policy to outperform CONWIP policy. Therefore, under a high process variation, the effect of WIP control dominates the effect of demand information flow. For hybrid policies, since all of them have kanban control, the tighter and more correlated WIP control in CK and ECK helps improve the performance over EK and GK policies in case 2.

4.6 Discussion

For traditional policies, CONWIP performs better than Kanban when demand and process variation are not greatly different. In opposite, Kanban performs better when the difference in variations is large. For hybrid policies, the performances of them generally fall between the best traditional policy and the Extended CONWIP Kanban policy.

One advantage of hybrid policies over traditional ones which was not explicitly shown in this comparison is that they are more versatile in achieving more possible choices of service levels. For instance, CONWIP policy had only seven possible choices of service levels between 90% - 99.99%, while the Extended CONWIP Kanban has many hundreds of possible choices. Therefore, it would be an unfair comparison if we tried to compare and optimize each policy against a given service level. However, this flexibility should not be neglected either.

We also did not consider the effect of having different holding costs of WIP for different locations. If this is the case, the hybrid policies which have more control on the WIP in between the line would exhibit more cost/service level tradeoffs. In addition, since we did not assign any penalty costs for the variability of WIP in the production line, the Base Stock and CONWIP policies which generally have higher variability might seem to perform better than it would be in reality.

4.7 Conclusion

In this chapter, we have studied and compared the performance of traditional and hybrid control policies in a variety of production situations. In all cases, the hybrid control policies demonstrated superior performance in achieving a high service level with minimal inventories compared with traditional ones. In addition, the Extended CONWIP Kanban policy dominates all other policies in all cases.

These results emphasize the importance and the robustness of ECK policy. It does not necessarily perform significantly better than the other policies for a given production situation but it is guaranteed to perform well under all circumstances.

CHAPTER 5

PERFORMANCE COMPARISON:
STATE SPACE REPRESENTATION APPROACH

We study the key characteristics in all production control mechanisms discussed in this thesis with a two stages tandem production system. This system allows us to analytically evaluate the tradeoffs between backorders, finished goods and work in process and compare them to optimal control policies.

First, we analyze the effect of the number of control points in a two-stage system. Another issue that we analyzed is the effect of having single or more control parameters per stage. Our comparisons shed further light on the desirable properties and shortcomings of a given pull control mechanism.

The outline of this chapter is as follows. In Section 5.1 and 5.2 we introduce the elements of the model and the corresponding control objective. This is followed by an analysis of optimal control in Section 5.3. A closed queueing network approximation is then provided. In Section 5.5, we analyze each of the single-stage control policies. A comparison of the single-stage control policies is discussed in Section 5.6. Two-stage control policies will then be analyzed in Section 5.7 Finally, comparison of each control policy is discussed in Section 5.8.

5.1 Elements and notations of a two-stage production system

Consider the production system in Figure 5-1. It consists of two production stages, $\mathcal{MP}_1$ and $\mathcal{MP}_2$, connected by an intermediate buffer whose level is nonnegative. The first stage composes of m_1 machines and the second stage composes of m_2 machines. The state of the system can be described by a state vector $\vec{e} = \{mp_1, mp_2, b, fgi, dem\}$ where mp_i, b, fgi, and dem are the number of parts in queue $\mathcal{MP}_i$, $\mathcal{B}$, $\mathcal{FGI}$ and $\mathcal{D}$ respectively.

Using the similar framework as Van Ryzin *et al*. (1993), Veatch and Wein (1994) and Karesmen and Dallery (2000), we defined the state of the system as a state vector $\vec{x} = (x_1, x_2)$, where x_1 denotes the sum of the amount of parts in this intermediate buffer and the amounts of part being processed in stage 2 while x_2 denotes the finished goods inventory level subtract any backordered demands. Thus, negative finished goods inventory level represents the amount of backordered demands, while the positive one represents the amount of physical finished parts. Using this definition, of the five parameters in $\vec{e}$, only mp_1 is not explicitly represented. However, mp_1 can be derived from x_1 and x_2 as we define the control function of each policy in the later sections.

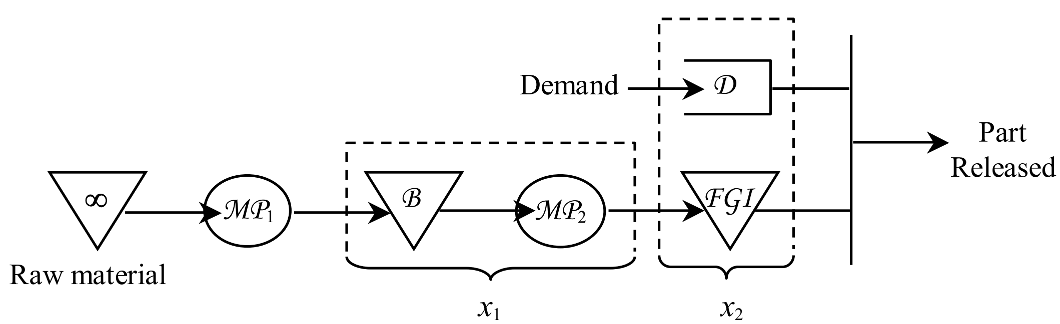


Figure 5-1: Two-stage production system

For the case where $\mathcal{MP}_1$ and $\mathcal{MP}_2$ compose of only one machine in each stage, the state space transition without any control boundaries is shown in Figure 5-2. We will define control boundaries in the next section. Note that μ_1, μ_2, and λ denote the production rate out of stage 1 and stage 2 and the demand arrival rate respectively.

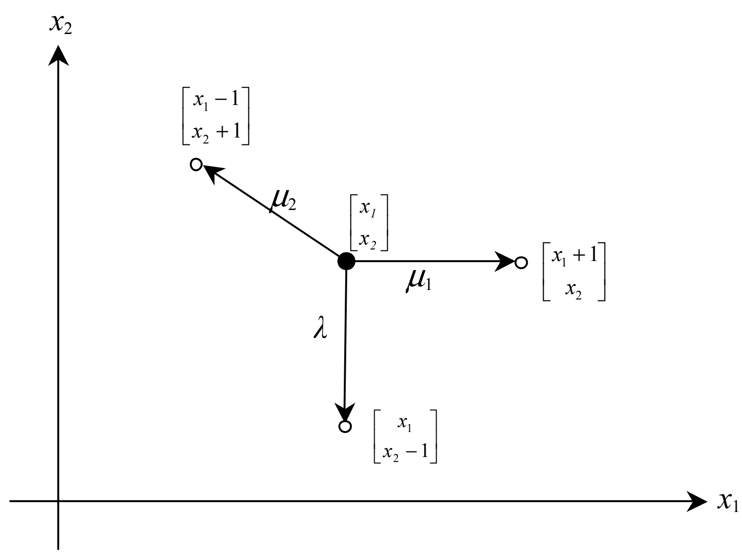


Figure 5-2: Stage-space transition diagram

5.2 Control objective

For the above system, we would like to minimize the cost of carrying inventory and shortage. We defined the piecewise linear cost function associated with the system as follow:

$$\text{Cost}(x_1, x_2) = c \cdot x_1 + h \cdot x_2^+ - b \cdot x_2^- \tag{5.1}$$

where $x_2^+ = \max\{0, x_2\}$ *and* $x_M^- = \min\{0, x_2\}$

We are interested in controlling the release of parts from the intermediate buffer to the downstream stage so that the sum of the long run average holding and backorder costs are minimized. A part release control policy, π, determines dynamically whether the machines in each stage should be authorized to work or not. Our objective is to find a part releasing policy π the minimize the long run average cost per unit time.

$$\limsup_{T\to\infty} \frac{E[\int_0^T c \cdot x_1(t) + h \cdot x_2^+(t) - b \cdot x_2^- \; dt]}{T} \tag{5.2}$$

5.3 Optimum control in the literature

It has been shown by Veatch and Wein (1994) that optimal control policies have certain monotonicity properties. Veatch and Wein (1994) showed that for both machines the "produce/do not produce" regions are separated by monotone switching curves. Figure 5-3, adapted from Veatch and Wein (1994), Van Ryzin, Lou, and Gershwin (1993) and Karaesmen and Dallery (2000), shows a typical switching curves and control region in the (x_1, x_2) state space.

The optimal control for stage 1 is to stop producing when the amount of work in process, x_1, is above the switching curve SC_1 even production is possible, otherwise stage 1 will keep producing. Note that SC_1 is decreasing in x_2. The optimal control for stage 2 is to stop producing when the finished goods inventory level, x_2, is above the switching curve SC_2, otherwise stage 2 will keep producing. SC_2 is non-decreasing in x_1. Therefore, the state space can be divided into three regions. In region B1, stage 1 produces, while stage 2 does not produce even it is possible. In region B2, stage 2 produces, while stage 1 does not produce. In region B3, no control action is exerted and both stages produce. For other region, neither stage operates.

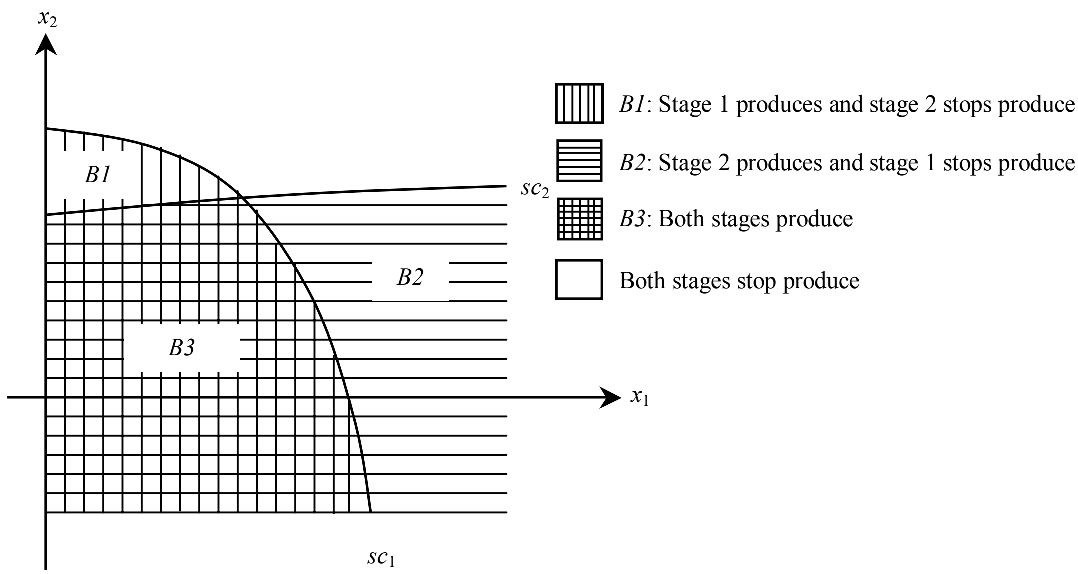


Figure 5-3: Monotonicity property of optimal policy.

The regions where only stage 1 or stage 2 is authorized to produce change one dimension only toward the corresponding switching curve boundary. When the state reaches the respective switching curve, both stages start production. Then the probability distribution is such that the system is more likely to move toward the hedging point of the system, i.e. the point where the two switching curves intersect. Perturbing the system from its hedging point will put it into a region that drives the system back toward this point.

From all examples tested in Van Ryzin, Lou, and Gershwin (1993), the control for stage 2 appeared to be a simple surplus-level control, i.e. machine 2 keeps producing until x_2 is greater than its hedging point value, i.e. the switching curve SC_2 is a horizontal line passing through the hedging point.

The control for the machine 1 works in a different way. When x_2 is below its hedging point position the optimal control is to produce more parts for the intermediate buffer, hence the increasing distance between SC_1 and the x_2-axis as x_2 decreasing. Intuitively, this larger buffer acts to increase the overall throughput of the system by reducing the probability that stage 2 will be starved of raw material to produce. However when x_2 is close to its hedging position, maximizing throughput becomes less important and minimizing the cost of buffer stock dominates. In the remainder of this chapter, we concentrate on characterizing the control regions for the case where both machines are operational, region B3 in Figure 5-3.

To summarize, the optimum policy can be characterized by two switching curves. The first is a horizontal line and the second is a concave monotone decreasing line. The intersection is called a hedging point. The feasible region is below both curves.

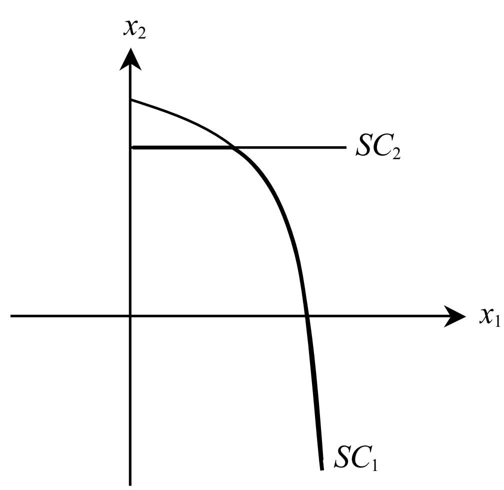


Figure 5-4: Optimal control policy

This optimal control policy is hard to find and harder to implement in practice, since it requires dynamic programming computation and is successful only for very simple systems. The optimal control policy can be closely approximated by a *two-boundary control policy* (Van Ryzin 1987), as shown in Figure 5-5. In Figure 5-5 we assumed that the solid line represents the optimal control policy, the dashed line represents the two-boundary control policy. The solid black dot represents the hedging point of the optimal policy while the gray dot on the dashed line represents the hedging point of the two-boundary policy.

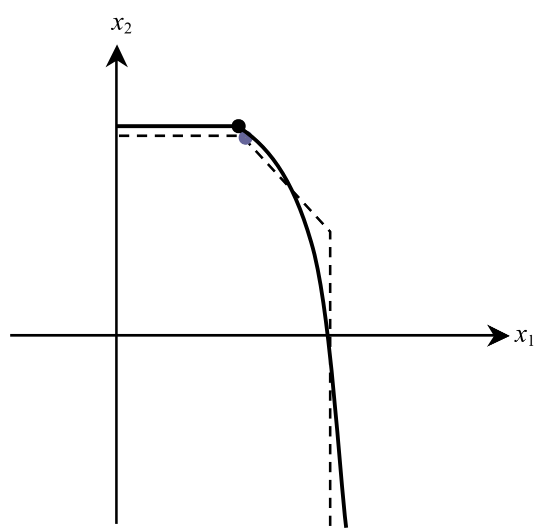


Figure 5-5: Two-boundary control policy

The two-boundary control approximates the optimal control by dividing the control region using two piecewise linear boundaries. The first boundary is characterized by assuming that stage 2 follows a pure surplus control, i.e. the horizon dashed line in Figure 5-5. For convenience, we call this horizontal line *finished goods limit* line. When the state of the system is on this boundary, the finished goods buffer is considered full and second stage stops working.

The second boundary is a piecewise linear boundary and characterized by assuming that stage 1 follows a surplus and buffer level controls, i.e. the combination of the 45 degree and the vertical dashed lines in Figure 5-5. When the state of the system is on the 45 degree line, stage 1 follows a constant surplus control, i.e. it maintains the amount of production surplus of stage 1, $x_1 + x_2$, to be constant. For convenience, we call the 45 degree line the *inventory limit* line since it limits sum of the work-in-process inventory, $x_{1,}$ and the finished goods inventory, x_2. When the state of the system is on the vertical boundary, the intermediate buffer is considered full and machine 1 stops working; hence we call the vertical line the *buffer limit* line since it limits the maximum amount of parts in the intermediate buffer.

Figure 5-6 (a) – (f) shows possible switching curves of two-stage boundary control under various production situations. See Van Ryzin, Lou, and Gershwin (1993) for more details. There are two important points in the graphs. First is the hedging point of the system which is the point connecting the finished goods limit and the inventory limit. It is denoted by a black dot in the figures. The second point is the turning point on SC_1 which connecting the inventory limit and the buffer limit. The locations of these two points are the most important characteristic of the system.

The location of the hedging point can be on the x_2-axis or in the first quadrant. It can be adjusted by moving the finished goods limit line or inventory limit line. The hedging point moves outward from the x_2-axis as the holding cost of finished goods compared with holding cost of WIP increases. Figure 5-6 (a) – (c) shows the cases where the hedging points are on the x_2-axis, while Figure 5-6 (d) – (f) shows the cases where they have moved out of the axis.

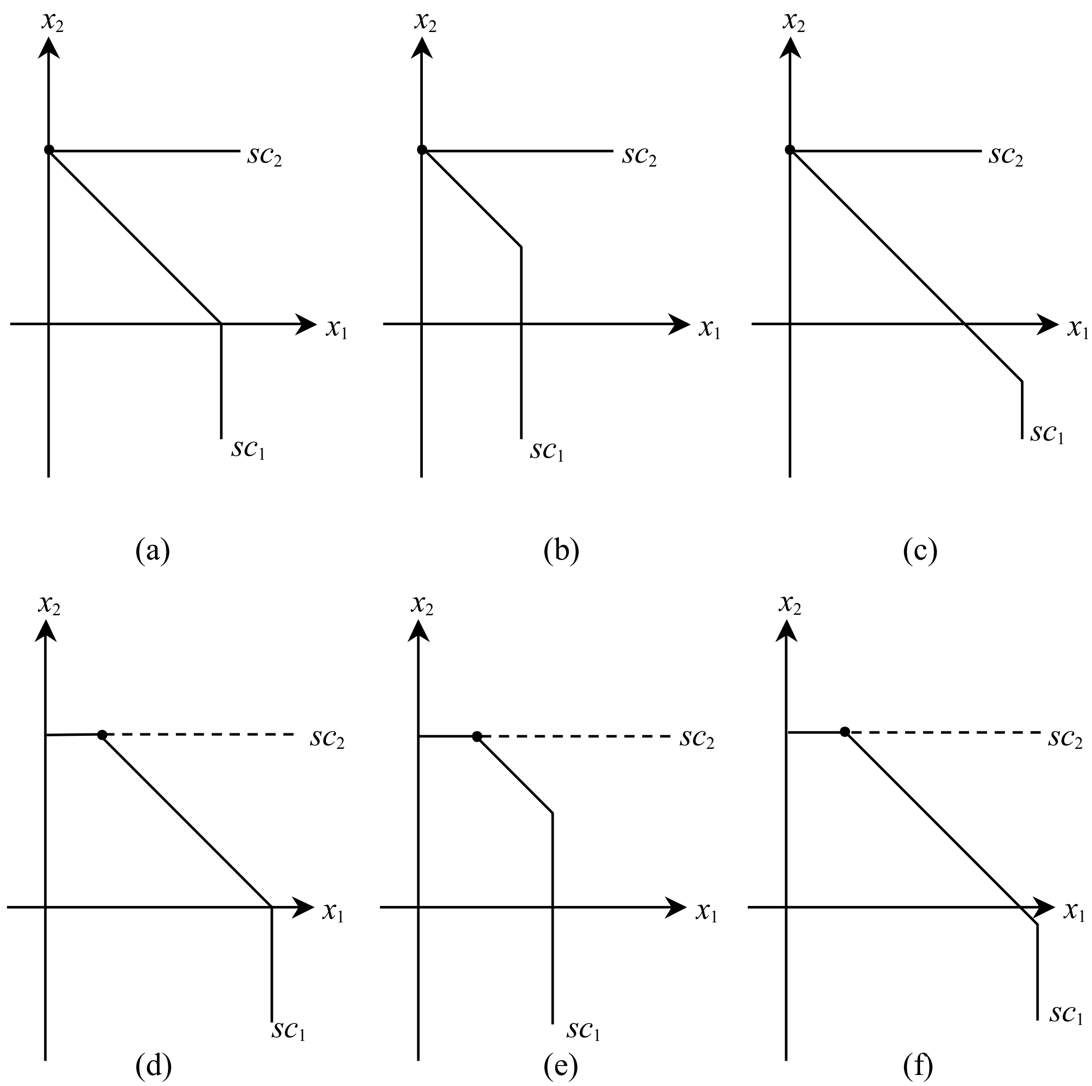


Figure 5-6: Examples of control regions of two-boundary control

For the locations of the turning point of SC_1, it can be above, below, or on the x_1-axis. Figure 5.6 (a) and (d) shows the cases where the turning point is on x_1-axis. Figure

5.6 (b) and (e) shows the cases where it in above the x_1-axis and Figure 5.6 (c) and (f) shows when it is below the x_1-axis. The location of this turning point can be adjusted by moving the buffer limit line or the inventory limit line. In general, this turning point moves upward as the production process is more reliable or has less variability and moves downward otherwise.

Based on the monotonicity properties of the optimal control and the structure of the two-boundary control policy, we analyzed the performance of different control policies by comparing them with the two-boundary control. The key insight is that a good control policy should be able to be configured such that the operating switching curves can be modified as close to the optimal switching curve as possible.

5.4 Closed queueing network approximation

The original queueing network of each control policy can be viewed as a multi-class closed queueing network with synchronization mechanisms. Each class of customers is a closed class corresponding to the number of parts of a particular stage. The principle of the method is to set the production rate of stage i, μ_i, equal to the load-dependent service rate of that stage, $\mu_i\ (n_i)$, where n_i is the amount of parts currently present at stage i.

Thus the production rates of each production stage in the original network are approximated by the production rate of that stage treated as a single-class queueing network and are independent from the number of parts of the other stage. This technique is tested to be successful for kanban and generalized kanban control systems (see Baynat et al 2001, Baynat and Dallery 1996)

For the case of a serial balance line with exponentially distributed processing times, it turns out that these load-dependent service rates can be obtained from the product-form solutions of the single-class network given by Hopp and Spearman (1996).

$$\mu(w) = \frac{w \cdot r_b}{w + w_0 - 1} \tag{5.3}$$

Where $\mu(w)$ = the production rate of the closed queueing network having w customers in the system

r_b = bottleneck rate

w_0 = critical WIP level, i.e. WIP level attained for a line with maximum throughput operating at the rate of the bottleneck.

Later in the analysis, the parameters in the numerical examples will have following values. Each stage consists of two identical machines in tandem and has *i.i.d.* exponential service time with rate 1. Demand is Poisson process with rate 0.5. Then, for each stage r_b = 1 and w_0 = 2 and the production rate out of each stage is computed using Equation (5.4)

$$\mu(w) = \frac{w}{w+1} \tag{5.4}$$

5.5 Single-stage control policies

In this section, we analyze the single-stage-controlled system. This single-stage control policy combines every manufacturing stage and treats them as a single control stage. This can also be considered as an input-output control policy. We will consider: CONWIP, single-point Base Stock, and single-point Extended CONWIP Kanban controls. In CONWIP and single-stage Base Stock policies, there is only one parameter to be specified, while there are two parameters to be specified in the single-point ECK policy. In CONWIP and single-stage Base Stock policies, there is only one parameter to be specified, while there are two parameters to be specified in the single-stage ECK policy.

We study the effect of controlling the amount of work-in-process in the CONWIP controlled system, the effect of setting the target finished goods level in the Base Stock controlled system, and finally, the effect of controlling both the amount of work-in-process and the target finished goods level in single-stage ECK controlled system. The purpose of this section is to capture the effects of each control actions and compare the benefit of having two control parameters versus one control parameter.

5.5.1 Control function of WIP-controlling policy

Figure 5-7 shows the queueing network model of the system under CONWIP control. Details dynamics of the system have already been discussed in section 2.1.

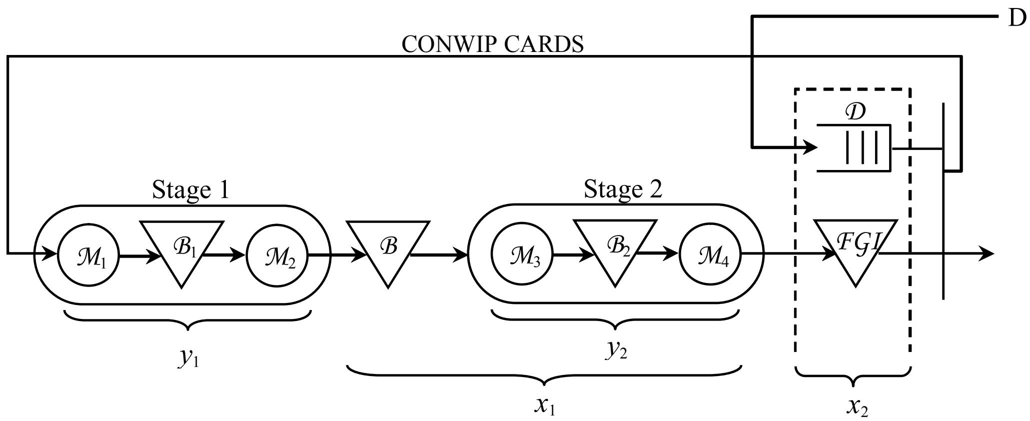


Figure 5-7: CONWIP control for two-stage four-machine production system

Here, we also introduce parameter y_1 and y_2 which represent the amount of parts in stage 1 and stage 2 respectively. These parameters will then be used to compute the production rate out of stage 1 and stage 2. Using the CONWIP invariant in Equation 2.1 described in Section 2.2, we have the following WIP relationship.

$$y_1 + x_1 + x_2^+ = C \tag{5.5}$$

From (5.5) and CONWIP bounds in property 3.12, the Inventory limit, buffer limit and finished goods limit can be expressed by the following relation ships:

$$x_1 + x_2 \le C \tag{5.6}$$

$$0 \le x_1 \le C \tag{5.7}$$

$$x_2 \le C \tag{5.8}$$

The production rates out of each stage can be approximated by the long-run closed queueing network approximation which depends only on the amount of parts in each stage, y_1 and y_2. From Equation (5.5), the production rate out of stage 1 can be approximated by:

$$\mu_1(y_1) \quad = \quad \mu_1(C - x_1 - x_2^+) \tag{5.9}$$

Since there will be no blocking at buffer $\mathcal{B}$, it is always empty, i.e. the finished part from stage 1 will proceed to stage 2 immediately and $x_1 = y_2$. Therefore, the production rate out of stage 2 can be approximated by;

$$\mu_2(y_2) \quad = \quad \mu_2(x_1) \tag{5.10}$$

Using the bounds and production rates defined in Equation (5.6) – (5.10), we can draw the state space transition rate diagram of the system as shown in Figure 5-8.

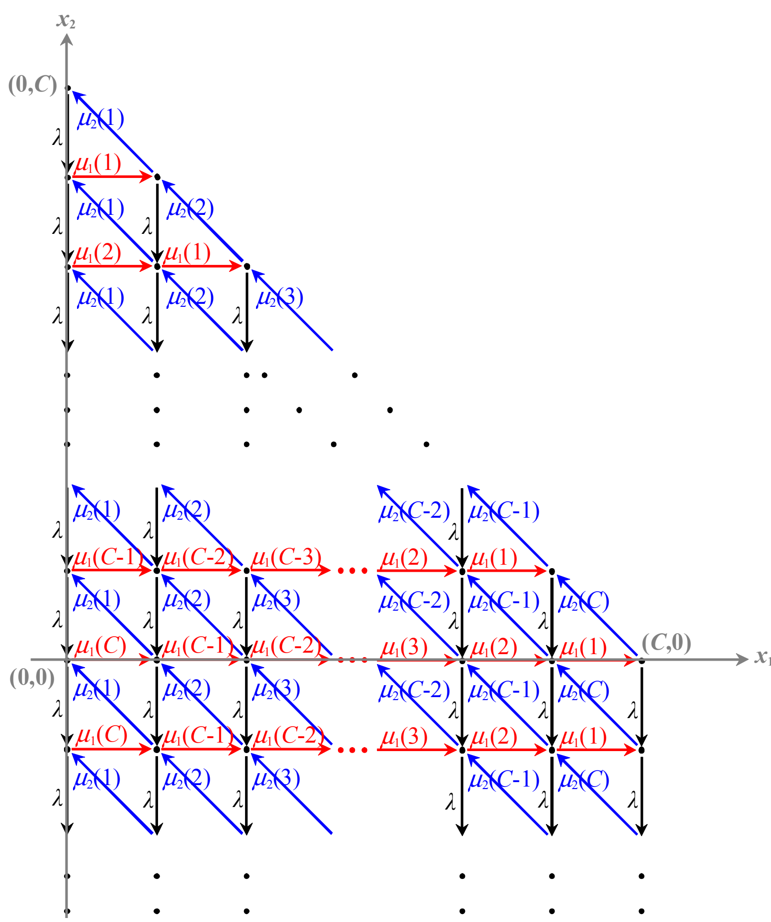


Figure 5-8: State-space transition rate diagram of CONWIP policy

Figure 5-9 shows the simplified version of Figure 5-8. The black solid line represents the transition boundary, i.e. the border of feasible state space. The solid red and blue lines represent the transition rate boundary of stage 1 or stage 2, at which the production rates reach its maximum. The dashed red and blue lines show the trend of the production rates once the state moves away from the boundaries. For example, the production rates of stage 1 of every states on the dashed red line shown in the figure are equal to $\mu_1(k_1\text{-}1)$. The transition rate of stage 1 decreases as the state moves to the right of the red boundary, while the transition rate of stage 2 decrease as the state moves to the left of the blue boundary. Note that the demand arrival rates, λ, are not shown in this simplified version since they are the same for all states.

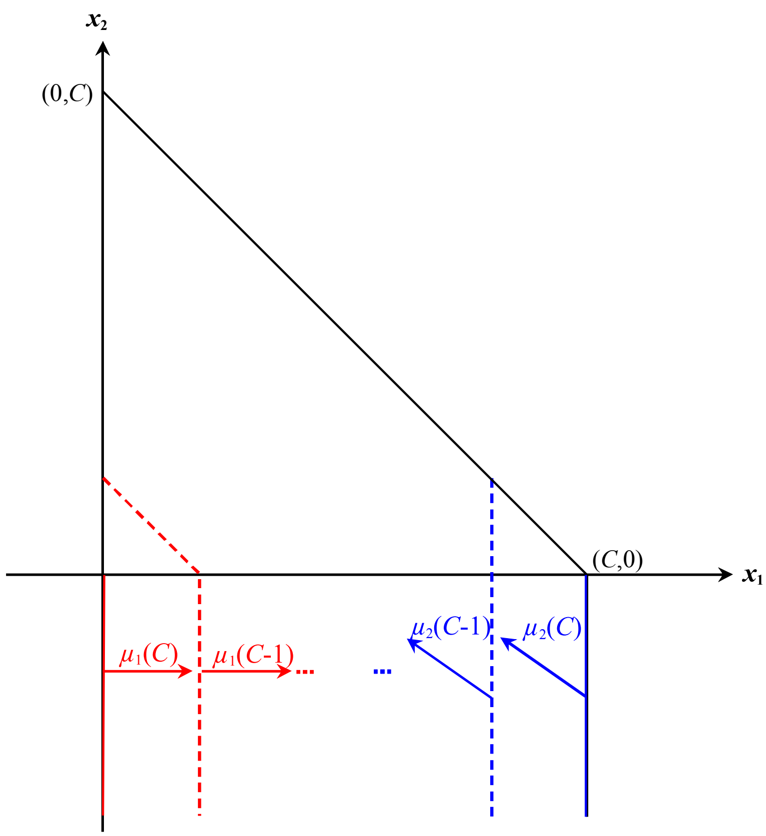


Figure 5-9: Simplified state-space transition rate diagram of CONWIP policy

At the transition rate boundary for stage 1, the red solid line, stage 1 contains C amount of parts, i.e. all of the allowable parts in the system, and the production rate of stage 1 reach its maximum at $\mu_1(C)$. Similarly, at the transition rate boundary of stage 2, stage 2 contains all allowable parts and the production rate of stage 2 reach its maximum at $\mu_2(C)$.

By comparing the state space of the CONWIP policy with that of the optimum, the switching curve for stage 2 is the horizontal line at $x_2 = C$. The switching curves for stage 1 constitute of a 45 degree line and a vertical line when $x_2 < 0$. As a result, the switching curves of stage 1 and stage 2 are dependent and can not be adjusted individually. Therefore, CONWIP policy works well only for the case where the optimal hedging point is at $x_1 = 0$ and the optimal switching curve for stage 1 changes its slope at the x_1-axis. This is similar to the state space of the two-boundary control shown in Figure 5-6 (a). The suitable production situation is when the finishing goods holding cost is not much larger than the WIP holding cost and process variation is not greatly different from demand variation.

The boundaries of the states which define the service level of the system, i.e. the area above the x_1-axis, is also defined by the CONWIP parameter. For convenience, we call these boundaries the *service level boundaries*. As a result, the transition boundaries, transition rate boundaries, and service level boundaries coincide and defined by the single parameter C. This causes inflexibility in the CONWIP policy to adjust the shape of its state space.

As an example, we compute the probability density at each state of this CONWIP system having the amount of CONWIP equals to 10 using the bounds and production rates in Equation (5.6) – (5.10). Figure 5-10 shows the graph of this probability density function.

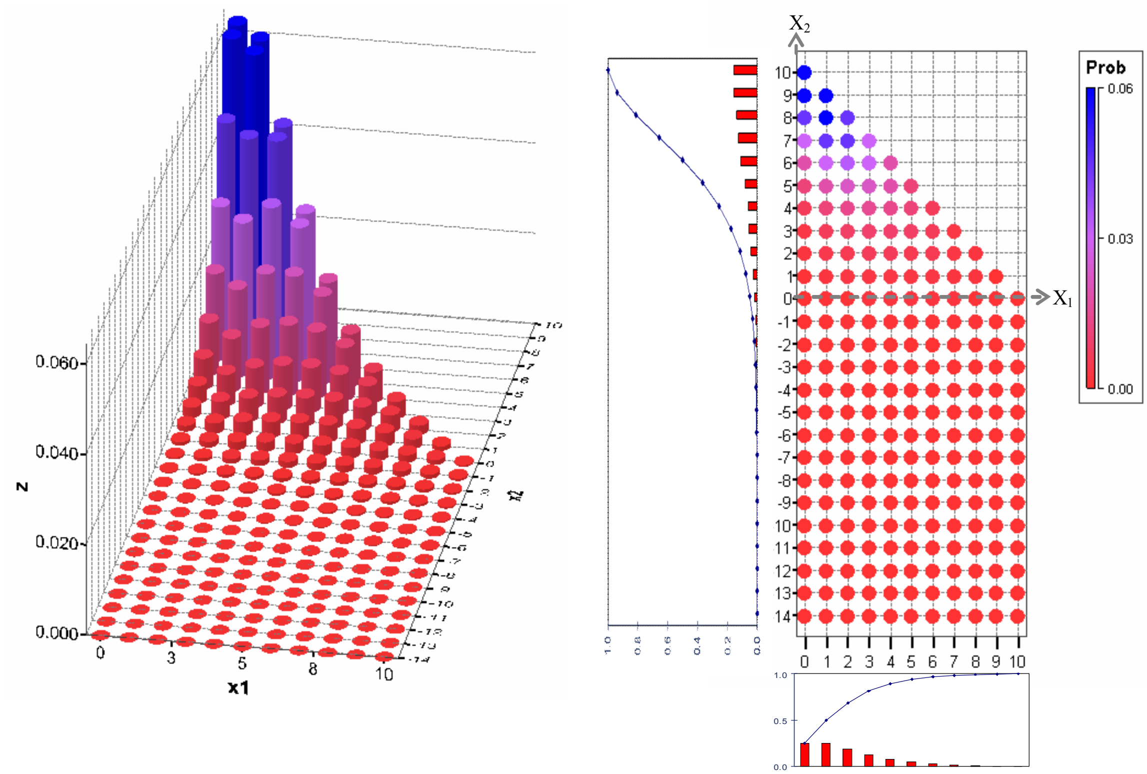


Figure 5-10: Probability density of CONWIP system (CONWIP = 10)

From Figure 5-10, the probability density of the system concentrates toward the hedging point, (0, *C*). From the cumulative distribution shown in the right side, the cumulative distributions of finished goods and WIP inventories are pushed toward the hedging point.

Figure 5-11 shows the state-space transition rate diagram when we adjust the amount of CONIWP, $C \rightarrow C + \Delta C$. The original transition boundaries are shown in dotted-dashed black line. The dotted-dashed red and blue lines represent original transition rate boundary of stage 1 and 2 respectively.

Here, the transition boundaries expand toward the right causing the hedging point of the system to move up from $(0, C)$ to $(0, C+\Delta C)$ and the transition rate boundary of stage 2 to move to the right. The maximum production rate of each stage is also increased to be $\mu_1(C+\Delta C)$ and $\mu_2(C+\Delta C)$. The expansion of state space above the x_1-axis increases the service level of the system. The upward movement of the hedging point causes increases in average finished goods holding cost. Therefore, increase in service level has to be sacrificed by increase in finished goods holding cost. As a result, in the production situation where required service level is very high and the finished goods holding cost is significantly greater than WIP holding cost, the CONWIP system is less desirable.

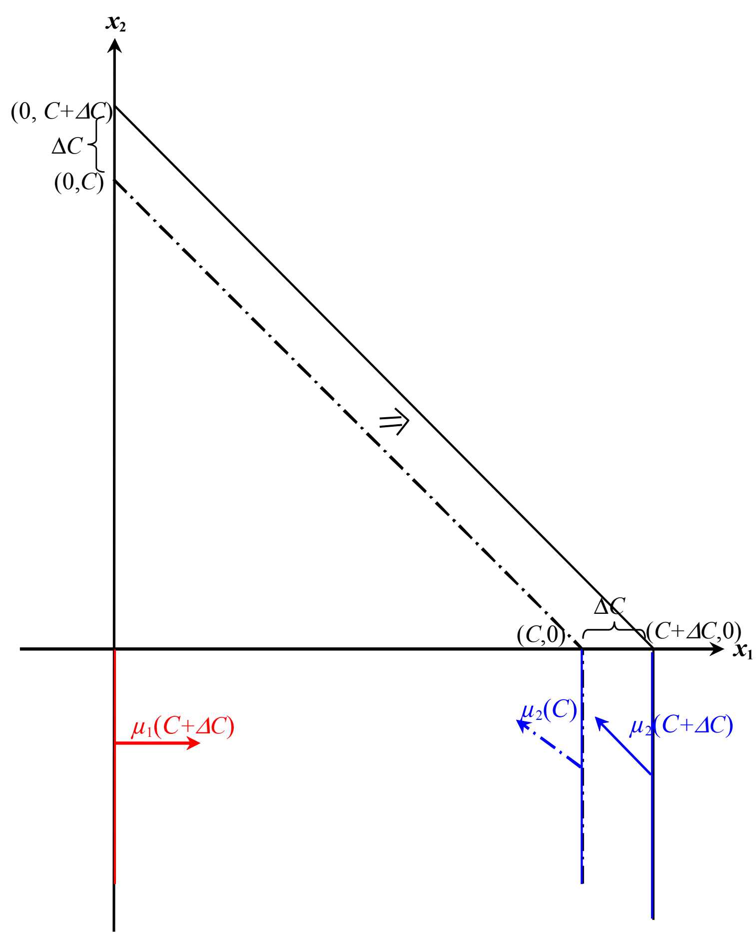


Figure 5-11: State-space of CONWIP policy when $C \rightarrow C + \Delta C$

5.5.2 Control function of finished goods basestock-controlling policy

The control rule of each stage is to keep producing parts until the finished goods inventory reach its basestock level. Figure 5-12 shows the queueing network model of this system. Details dynamics of the system have been discussed in section 2.2.

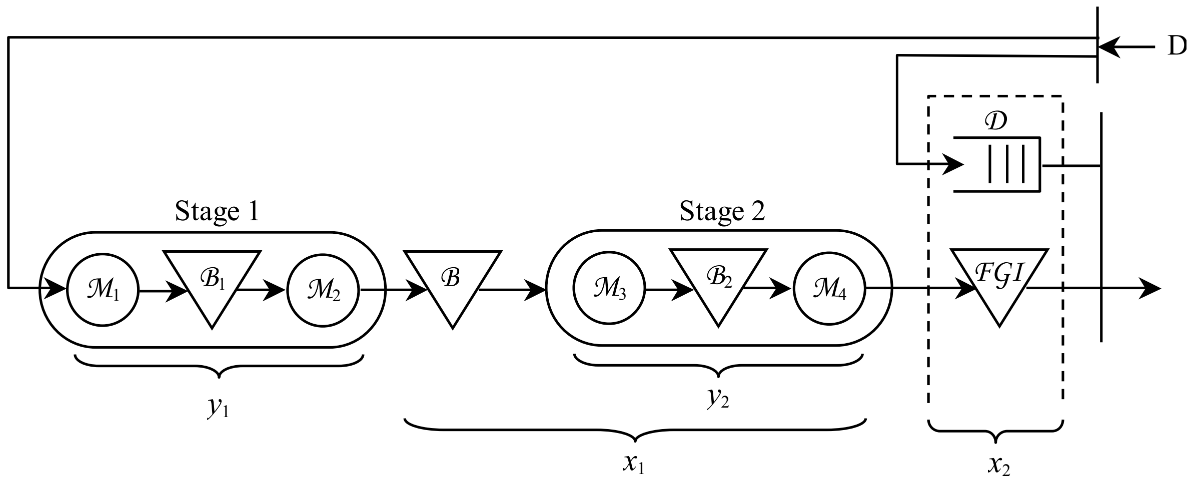


Figure 5-12: Base Stock control for two-stage four-machine production system

Using invariant of basestock r form Equation 2.2 in Chapter, the WIP relationship of the system is characterized by the following equation.

$$y_1 + x_1 + x_2 = s_2 \tag{5.11}$$

Using the WIP relationship, the buffer limit and finished goods limit can be expressed by the following relation ships:

$$x_1 + x_2 \leq s_2 \tag{5.12}$$

$$x_2 \leq s_2 \tag{5.13}$$

The production rate out of stage 1 can be approximated by

$$\mu_1(y_1) \quad = \quad \mu_1[s-(x_1+x_2)] \tag{5.14}$$

Since there will be no blocking at buffer $\mathcal{B}$, $x_1 = y_2$. The production rate out of stage 2 can be approximated by;

$$\mu_2(y_2) \quad = \quad \mu_2(x_1) \tag{5.15}$$

Using the bounds and production rates defined in Equation (5.12) – (5.15), we can draw the state space transition rate diagram of the system which is shown in Figure 5-13.

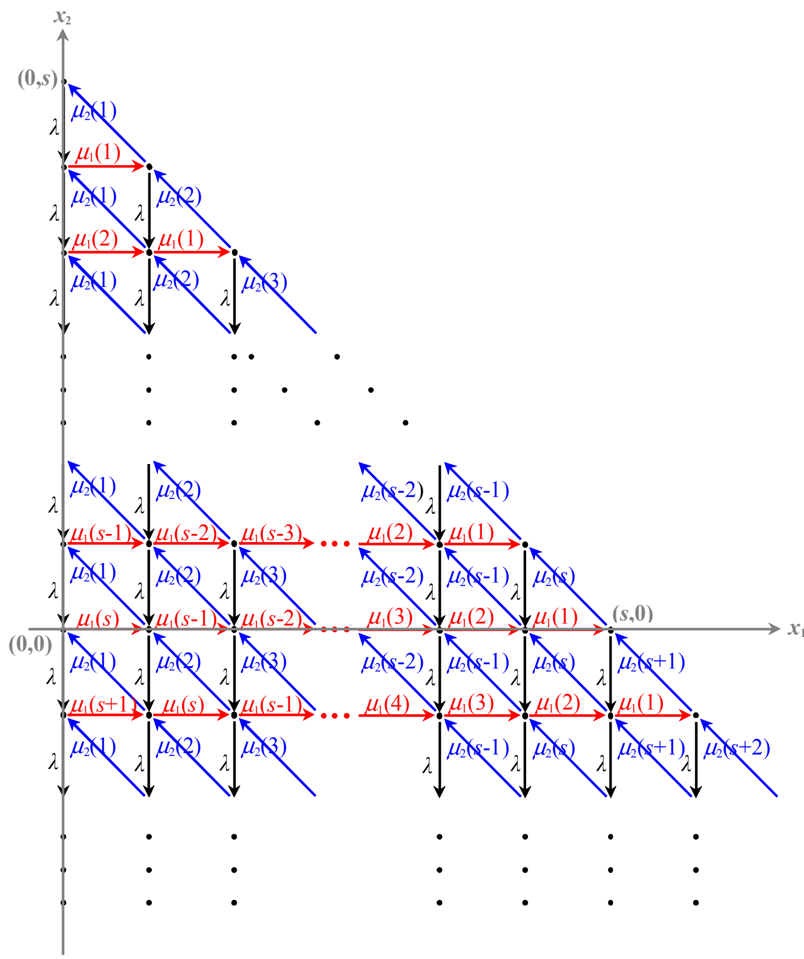


Figure 5-13: State-space transition rate diagram of single-stage Base Stock policy

Figure 5-14 shows the simplified version of the state space transition rate diagram. Since there is no transition rate boundary in this control policy, we shown the trend of the transition rates by dashed lines. Again the production rate of stage 1 decreases as the state moves to the right and the production rate of stage 2 decreases as the state moves to the left.

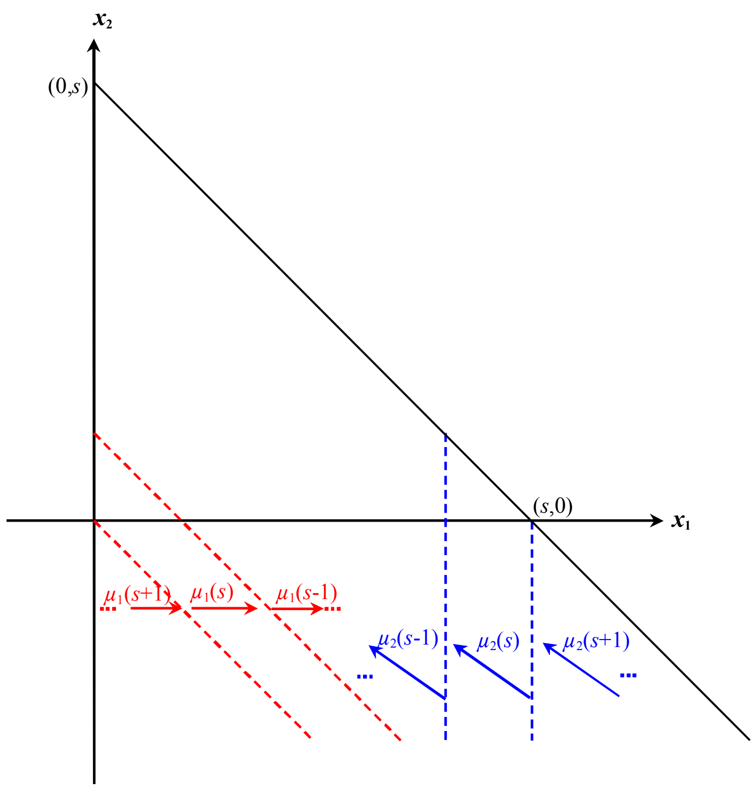


Figure 5-14: Simplified state-space diagram of single-stage Base Stock policy

The boundary of the state space is defined by parameter s and the shape of the state space is similar to the state space of the CONWIP policy except that there is no buffer limit line, i.e. no limit on the maximum amount of x_1. The service level boundary is defined by the same parameter s. The hedging point of the system is at its initial point (0, s) which is at the top left corner similar to the CONWIP policy.

Since there is no WIP control mechanism, there is no transition rate boundary in this control policy and the maximum production rate of stage 1 and 2 are $\mu_1(\infty)$ and $\mu_2(\infty)$.

By comparing the state space of the Base Stock policy with the optimal policy, this unbounded WIP level implies that the Base Stock policy is never optimum. Similar result can be found in Veatch and Wein (1994).

As an example, we compute the probability density at each stage of this Base Stock system having the amount of basestock equals to 10 using the bounds and production rates in Equation (5.12) – (5.15). Figure 5-15 shows the graph of this probability density function. Note that the picture did not show all possible states for $x_1 > 15$.

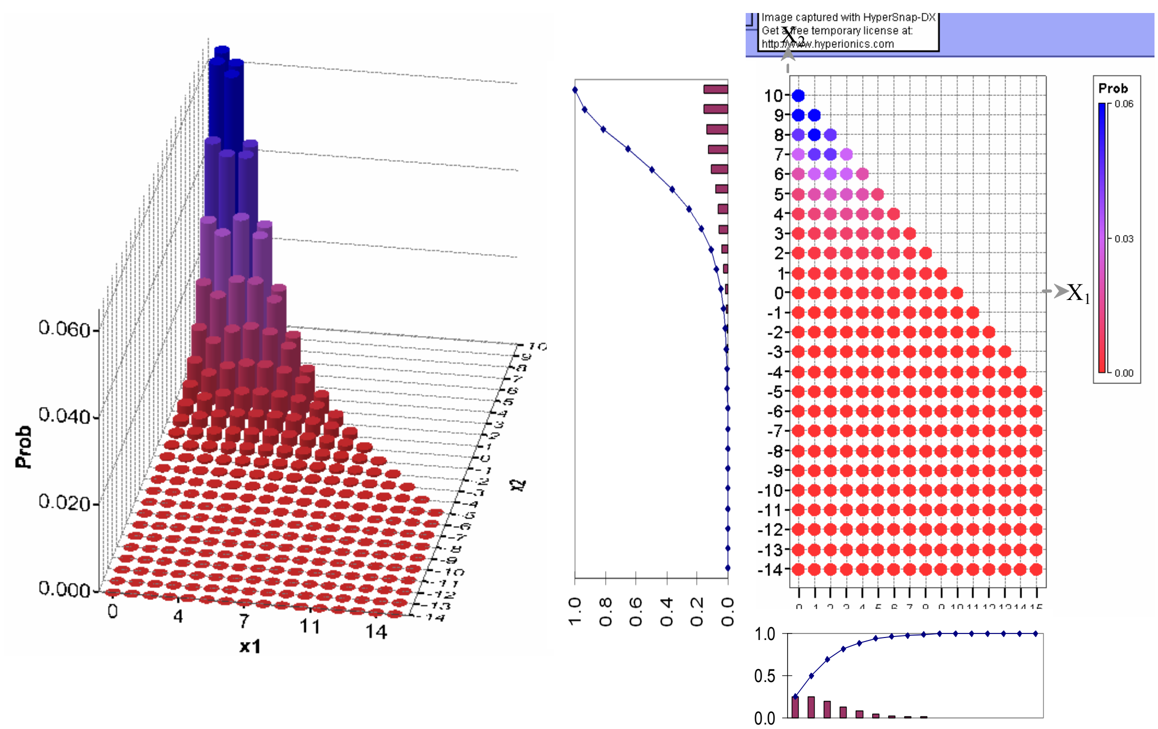


Figure 5-15: Probability density of single-stage Base Stock system (basestock = 10)

From Figure 5-15, it can be seen that the probability density of single-stage Base Stock system is similar to that of the CONWIP system except that the feasible state expand to the right infinitely, which causes the Base Stock system to have higher average WIP than a CONWIP system having $C = s$. Also, the cumulative distributions of finished goods and WIP inventories are pushed toward the hedging point.

By adjusting the basestock parameter, the transition boundary expands to the right as is the case in the CONWIP system.

5.5.3 Control function of single-stage ECK control policy

Single-stage ECK control can be viewed as a combination of single-stage Base Stock and single-stage CONWIP controls. The Kanban control mechanism of the ECK policy in this control policy is implicit since the CONWIP cards act the same as the kanban cards in this case. It requires two parameters to control the entire production line. The first parameter is the basestock level, s, which specifies the target finished goods level. The second parameter is the number of CONWIP cards C, which limits the maximum amount of work-in-process allowed in the production system. Figure 5-16 shows the queueing network model of a single-stage ECK control applied to a two-stage production line.

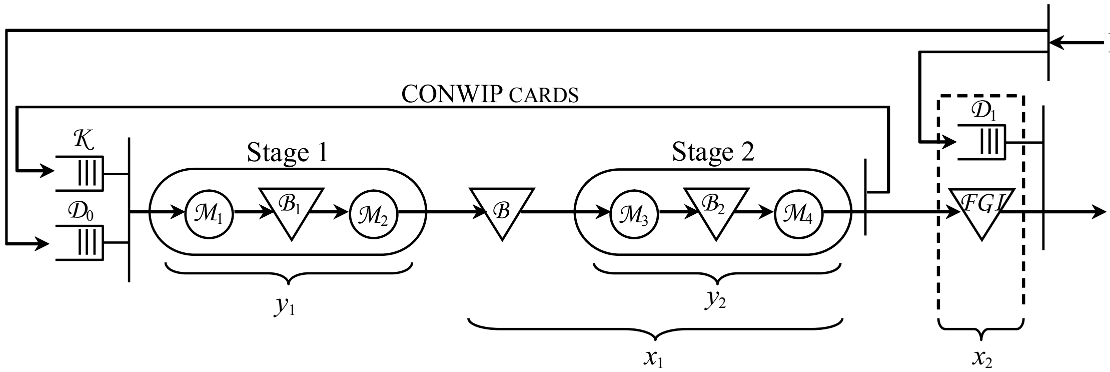


Figure 5-16: Single-stage ECK control for two-stage four-machine production system

Based on the Inventory limit and basestock mechanisms of the control mechanism, the limit boundaries of each stage can be expressed as follow.

$$x_1 + x_2 \leq s \tag{5.16}$$

$$0 \leq x_1 \leq C \tag{5.17}$$

$$x_2 \leq s \tag{5.18}$$

If the finished goods inventory position, x_2, does not fall C units below the target basestock level, the control policy for stage 1 is the same as in the single-stage Base Stock policy. Once the finished goods inventory falls below this point, there will be no further release of new raw material into the production line and the amount of parts in stage 1 is equal to the total number of CONWIP subtracts the amount of parts in stage 2. The production rate out of stage 1 will be approximated as follow:

$$\mu_1(y_1) = \begin{cases} \mu_1[s-(x_1+x_2)] & : \quad x_2 \geq s-C \\ \mu_1(C-x_1) & : \quad x_2 < s-C \end{cases} \tag{5.19}$$

Since there will be no blocking at buffer $\mathcal{B}$, the production rate out of stage 2 can be approximated as follow.

$$\mu_2(y_2) \quad = \quad \mu_2(x_1) \tag{5.20}$$

Figure 5-17 shows the simplified version of the state space transition rate diagram. It can be viewed as the state space of the single-stage CONWIP system having the x_1-axis moving up or down according to the s value.

The transition boundary is defined by parameter C, but the service level boundary can be adjusted by parameter s. By comparing the state space of the single-stage ECK policy with the optimal switching curve, the switching curve for stage 2 is implicit and is the horizontal line at $x_2 = s$. On the other hand, the switching curve for stage 1 changes its slope from -1 to ∞ when $x_2 = s - C$ and $x_1 = C$. Therefore, single-stage ECK policy works well only for the case where the optimal hedging point is at $x_1 = 0$. This is similar to the state space of the two-boundary control shown in Figure 5-6 (a)-(c). The suitable production situation is when the finishing goods holding cost is not much larger than the WIP holding cost.

Note that the state space diagrams shown in Figure 5-17 is for the case where $s < C$. When $s > C$, the switching curve boundary changes its slope above the x_1-axis. There are two special cases: When $s = C$, the state space of the system will be exactly the same as the one for the CONWIP system having C amount of CONWIPs and when $C \rightarrow \infty$, the state space of the system will be the same as the one for the single-stage Base Stock system having it target basestock of s units

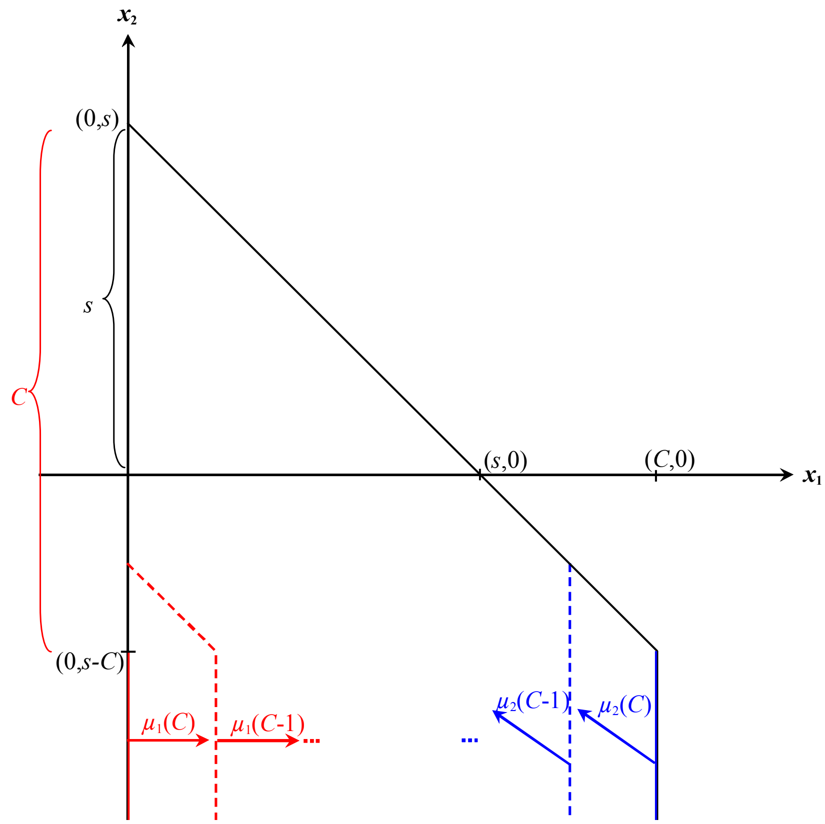


Figure 5-17: State-space transition rate diagram of single-stage ECK policy

The transition rate boundary of stage 1 is at $x_1 = 0$ and $x_2 \leq s - C$ with the maximum production rate of stage 1 equals to $\mu_1(C)$. The transition rate boundary of stage 2 is at $x_1 = C$ and $x_2 \leq s - C$ with the maximum production rate of stage 2 equals to $\mu_2(C)$.

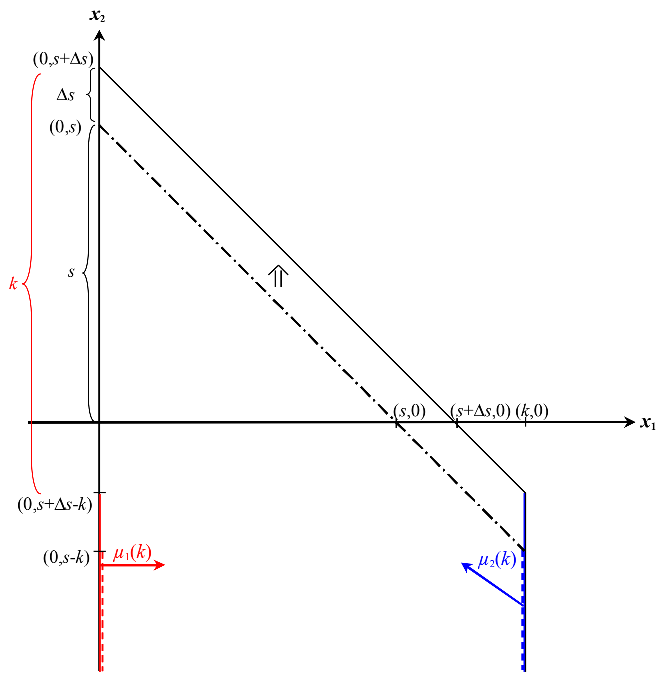
As a result, the transition boundary and the service level boundary can be independently specified. However, the transition boundary and the transition rate boundary are still coupled together and the maximum production rates of each stage are also defined by the same parameter.

Figure 5-18 shows the changes in the state space transition rate diagram when we change s to $s + \Delta s$ (Figure 5-17a) and C to $C+\Delta C$ (Figure 5-17b). Note that in Figure 5-

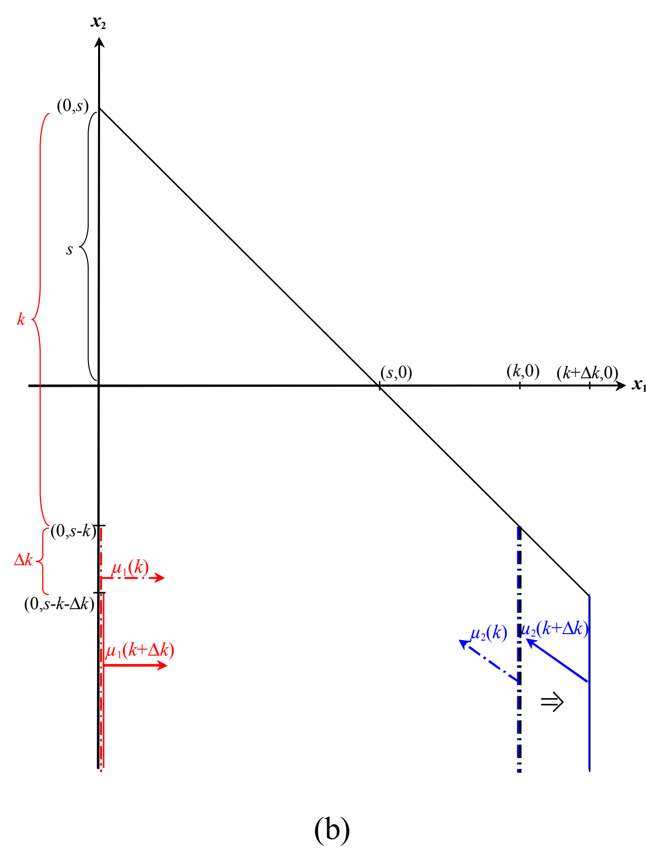
18, we showed only the case where $s < C$. The dotted-dashed black line represents the original state space boundaries. The dotted-dashed red and blue line represents the original transition rate boundaries of stage 1 and 2 respectively.

For changes in s, assuming that the amount of CONWIPs is held constant, the number of reachable states and the probability density at each state remains the same as before the changes while the number of state above the x_1-axis changes, i.e. only the service level changes while the average WIP level which depends on x_1 remains the same. Here, the transition boundaries expand upward causing the hedging point of the system to move up from $(0, s)$ to $(0, C+\Delta s)$ and the transition rate boundary of stage 1 and stage 2 also move upward. However, the maximum production rates of each stage remain the same. It also causes the turning point of stage-1 switch curve to move in the same direction as the hedging point.

The advantage of this adjustment is that one can adjust the service level of the system through the adjustment of finished goods inventory without changes in WIP level. For the limit on the s value, the minimal amount of s is zero and the maximum of amount of s is infinite.



(a)



(b)

Figure 5-18: State-space of single-stage ECK when parameters change

For changes in C, see Figure 5-18b, the numbers of possible states expand to the right if C increases or shrink to the left if C decreases. Therefore, the average WIP level changes according to the changes in C. Moreover, the probability density of each state also changes because of the changes in the number of states and the maximum production rates. We found that increasing the amount of CONWIPs beyond a certain point will almost give no effect to the probability density of the states of the system because the probabilities of being in those new states are too small to make any significant changes to the probability density of the other states.

As a result, the optimal configuration for this control policy is to find the minimal s such that the service level constraint is satisfied given that the CONWIP limit is very large. Then try to decrease the amount of CONWIP limit which will make only slightly change in the service level of the system until the service level is violated. Similar analytical result can be found in Dallery and Liberlopoulos (2002).

5.6 Discussion on single-stage control policies

In Table 5.1, we summarized the effect of control parameters in each control policy in term of the transition boundaries and the production rate. It can be seen that for all single-stage control policies the finished goods limit and the Inventory limit are coupled together, i.e. defined by the same parameter, which cause the hedging point of these policies to be on the x_2-axis.

For the CONWIP policy, since the Inventory limit and buffer limit are also defined by the same parameter, the turning point of the switching curve of stage 1 is required to be on the x_1-axis. On the other hand, for the single-stage ECK, since the Inventory limit and the buffer limit are defined by separate parameter; hence, the turning point of the switching curve of stage-1 can be adjusted to be either above, below or on the x_1-axis.

Table 5.1: Comparison of single-stage control policies

Single-stage control Policies	Transition Boundaries			Production rate	
	FGI limit	WIP limit	Buffer limit	Stage 1	Stage 2
CONWIP	*C*	*C*	*C*	*C*	*C*
Single-stage Base Stock	*s*	*s*	-	-	-
Single-stage ECK	*s*	*s*	*C*	*s, C*	-

5.7 Two-stage control policies

In this section we study the effect of adding another control point in the system - the intermediate buffer between the two production stages. The control policies that we considered are Base Stock, Kanban, CONWIP-Kanban, Extended Kanban, Generalized Kanban, and Extended CONWIP Kanban control policies. In Base Stock, Kanban, and CONWIP-Kanban, there is only one control parameter at each control point, while there are two control parameters at each control point for Extended Kanban, Generalized Kanban, and Extended CONWIP Kanban. For CONWIP-Kanban and Extended CONWIP Kanban, there is an additional control parameter required for the entire production line. Details of each of these control policies can be found in Chapter 2

The purpose of this section is to capture the effects of each control actions and compare the benefit of having two control parameters versus one control parameter at each control point and the effect of adding the CONWIP control mechanism to form a hybrid control.

5.7.1 Control function of two-stage Base Stock policy

In this section we analyze a two-stage production system operating under Base Stock control. The control mechanism of this system is similar to the single-point Base Stock control we analyze in previous section; however, the intermediate buffer, $\mathcal{B}$, is now considered as part of the control mechanism. Figure 5-19 shows the queueing network model of a two-stage Base Stock control system.

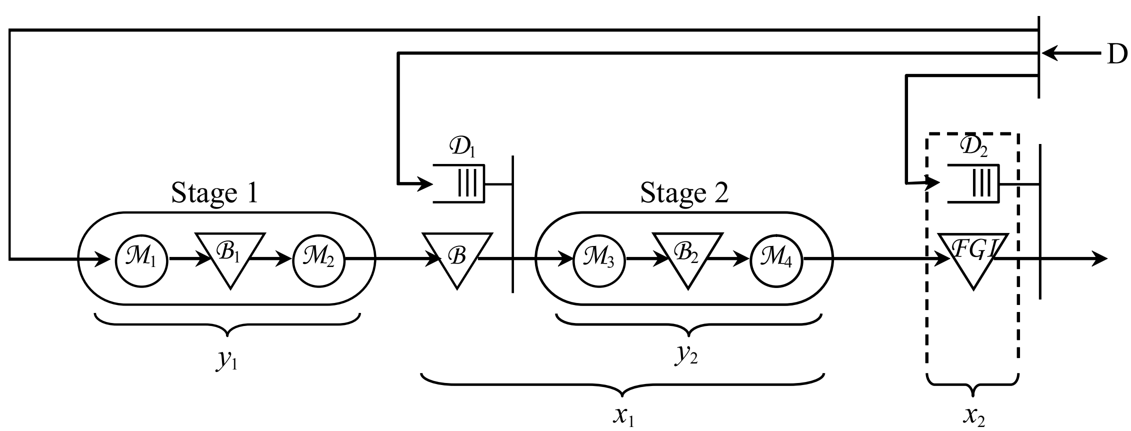


Figure 5-19: Two-stage Base Stock control for two-stage production system

The limit boundaries and the production rates of each stage can be expressed as follow (see Appendix A for details).

$$x_1 + x_2 \leq s_1 + s_2 \tag{5.21}$$

$$x_2 \leq s_2 \tag{5.22}$$

$$\mu_1(y_1) = \mu_1[s_1 + s_2 - (x_1 + x_2)] \tag{5.23}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(s_2 - x_2) & ;x_1 + x_2 \geq s_2 \\ \mu_2(x_1) & ;x_1 + x_2 < s_2 \end{cases} \tag{5.24}$$

Figure 5-20 shows the state space transition rate diagram of the system using the bounds and production rates defined above. The dashed red and blue lines represent the production rate of stage 1 and stage 2 respectively.

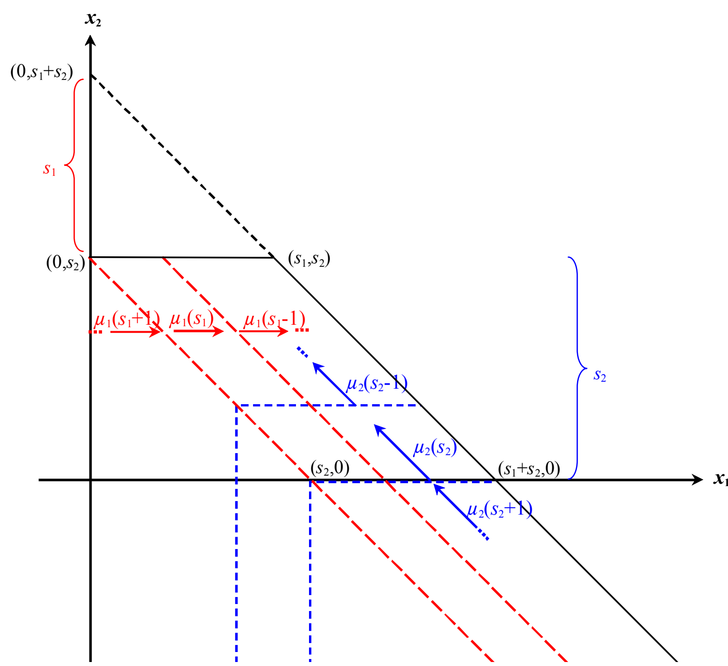


Figure 5-20: State-space transition rate diagram of two-stage Base Stock system

From Figure 5-20, the shape of the state space in the first quadrant is adjustable toward that of the optimum, i.e. the finished goods limit can be independently specified by parameter s_2. However, there is still no buffer limit in the system, i.e. $x_1 \rightarrow \infty$ when $x_2 \rightarrow -\infty$, which causes the Base Stock policy to never be optimum. Note that this system will be the same as single-stage Base Stock system if $s_1 = 0$.

Figure 5-21 show probability density of each state for the two-stage Base Stock system having $s_1 = 5$ and $s_2 = 7$.

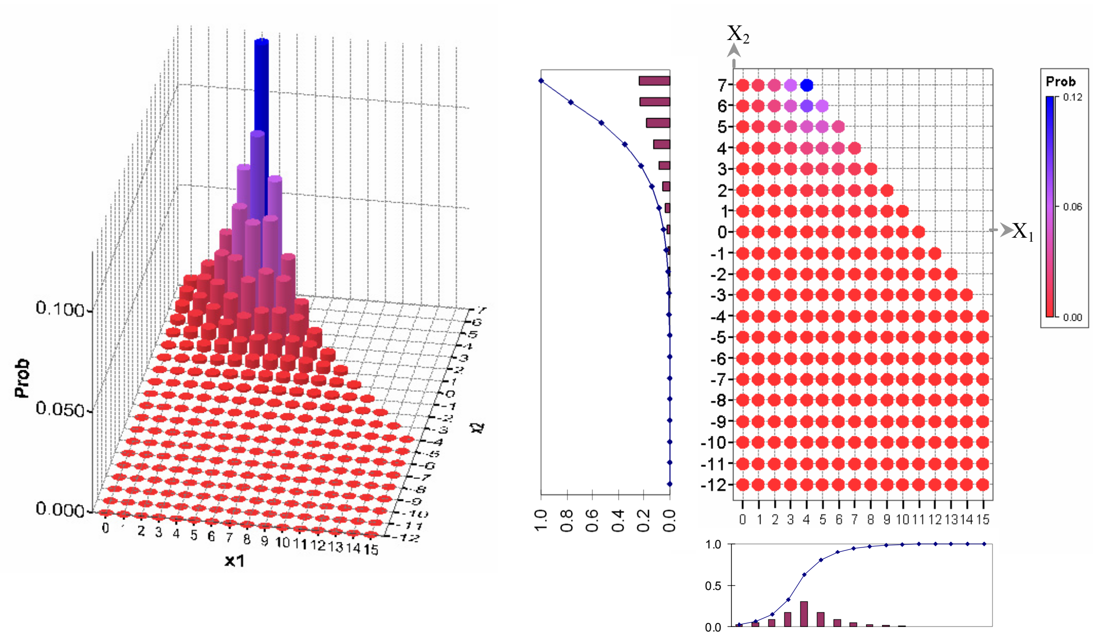
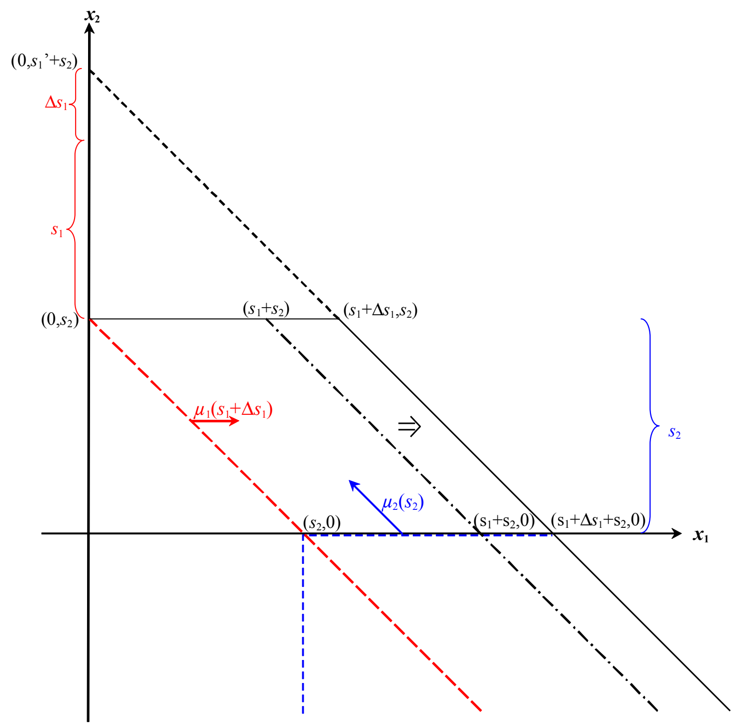


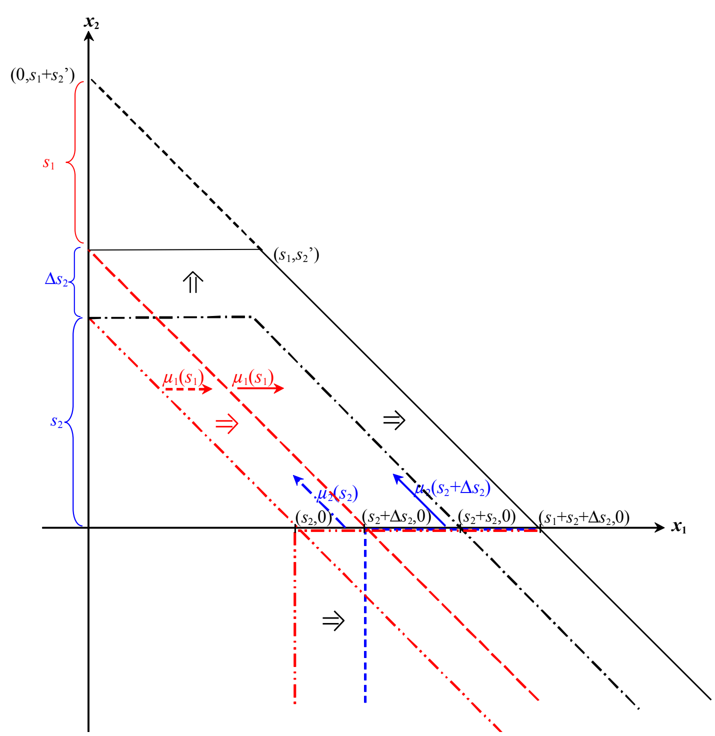
Figure 5-21: Probability density of two-stage Base Stock system ($s_1 = 5$ and $s_2 = 7$)

From Figure 5-21, it can be seen that the probability density of the system is peak at the hedging point of the system, (s_1, s_2), and then gradually reduces outward. By comparing this probability density with that of the single-stage Base Stock system, the average finished goods level is reduced by the increase in the average WIP level while the service level can remain the same.

Figure 5-22 shows the changes in the state space transition rate diagram when we change s_1 to $s_1 + \Delta s_1$ (Figure 5-22a) and s_2 to $s_{2} + \Delta s_2$ (Figure 5-22b).



(a)



(b)

Figure 5-22: State-space of two-stage Base Stock system when parameters change

As s_1 increases, the number of states above the x_1-axis increase, i.e. more states expand to the right and at the same time the hedging point of the system moves to the right. Thus, the service level and the amount of WIP of the system increase. As s_1 keeps increasing, the service level of the system converges to the service level of a production system containing only production stage 2 and its finished goods buffer because the probability that stage 2 will starve of parts to process converge to zero and hence depends only on the amount of s_2. Therefore, for a given value of s_2 the service level of the system is a concave function of s_1, i.e. the incremental increase in service level decrease as s_1 increase. On the other hand, the average WIP level increases as s_1 increase with out any bound.

For changes in s_2, as s_2 increases, the numbers of states above the x_1-axis increase while the hedging point of the system moves upward, i.e. the hedging point of the system remains at $x_1 = s_1$. As a result, the service level and average amount of finished goods increase while the average amount of WIP remains the same. The increase (or decrease) in s_2 can be viewed as the downward (or upward) shift of the x_1-axis in the original state space diagram; hence the cumulative distribution of x_1 remains the same which results in the same average WIP level.

5.7.2 Control function of two-stage Kanban policy

Kanban policy requires only one parameter per stage, which is the amount of kanbans of each stage k_1 and k_2. Figure 5-23 shows the queueing network model of a two-stage Kanban control system. See section 2.3 for detail dynamics of the system.

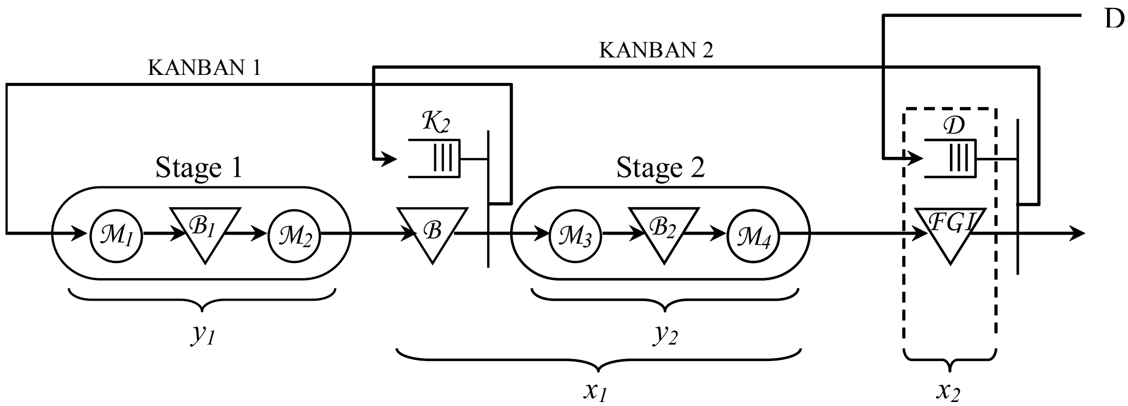


Figure 5-23: Two-stage Kanban control for two-stage four-machine production system

The limit boundaries and the production rates of each stage can be expressed as follow (see Appendix A for details).

$$x_1 + x_2 \le k_1 + k_2 \tag{5.25}$$

$$0 \le x_1 \le k_1 + k_2 \tag{5.26}$$

$$x_2 \le k_2 \tag{5.27}$$

$$\mu_1(y_1) = \begin{cases} \mu_1[k_1 + k_2 - (x_1 + x_2)] & ; x_1 + x_2 \ge k_2,\, x_2 < 0 \\ \mu_1[k_1] & ; x_1 + x_2 < k_2,\, x_2 < 0 \end{cases} \tag{5.28}$$

$$\mu_1(y_1) = \begin{cases} \mu_1[k_1 + k_2 - x_1] & ; x_1 \ge k_2,\, x_2 < 0 \\ \mu_1[k_1] & ; x_1 < k_2,\, x_2 < 0 \end{cases} \tag{5.29}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2 - x_2) & ; x_1 + x_2 \ge k_2,\, x_2 \ge 0 \\ \mu_2(x_1) & ; x_1 + x_2 < k_2,\, x_2 \ge 0 \end{cases} \tag{5.30}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ; x_1 \ge k_2,\, x_2 < 0 \\ \mu_2(x_1) & ; x_1 < k_2,\, x_2 < 0 \end{cases} \tag{5.31}$$

Using the bounds and production rate approximations defined above, the state space transition diagram is shown in Figure 5-24. The transition rate for states in the shaded area on the left of the transition rate boundary (red solid line) is equal to maximum production rate of stage one, $\mu_1(k_1)$, and the transition rate for states in the shaded area on the right of the transition rate boundary (blue solid line) is equal to the maximum production rate of stage 2, $\mu_1(k_2)$. The trends of transition rates are given in the dashed red and blue line.

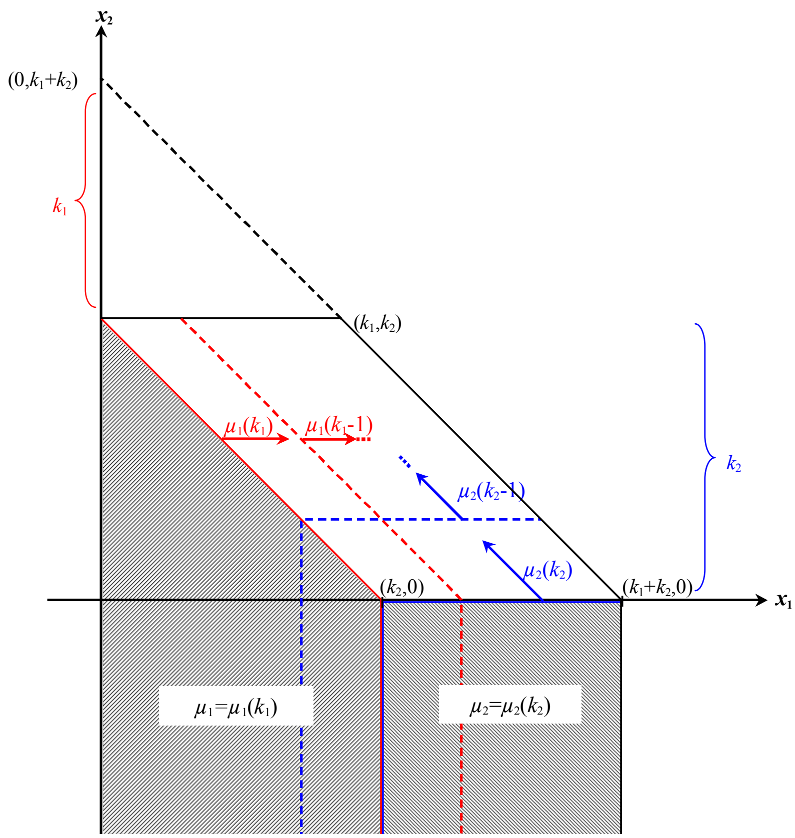


Figure 5-24: State space transition rate diagram of two-stage Kanban control policy

The shape of the state space of the two-stage Kanban policy is the same as that of the two-point Base Stock policy in the first quadrant. However, in the fourth quadrant, there is a constant buffer limit on x_1, $k_1 + k_2$.

By comparing the shape of the state space of the Kanban policy to that of the optimum, the state space of Kanban system be adjust to be similar to those in Figure 5-6

(a) and (d). This is because the touching point between the Inventory limit and the buffer limit is required to be on the x_1-axis, i.e. the switching curve of stage 1 has to change its slope at the x_1-axis.

Once the transition boundaries are specified, the transition rate boundaries can no longer be adjusted, i.e. the transition rate boundaries are defined by the same parameter as the transition boundaries. The transition rate boundaries of each stage are joined at $x_1 = k_2$.

As an example, we compute the probability density at each state of for the two-stage Kanban system having $k_1 = 6$ and $k_2 = 7$ using the bounds and production rate in Equation (5.25) – (5.31). Figure 5-25 shows of the graph of this probability density function.

It can be seen that the distribution of the probability density of the Kanban system is similar to that of the Base Stock system, i.e. it peaks at the hedging point of the system, (k_1, k_2), and then gradually reduces outward. However, since there is a buffer limit here, the cumulative distribution of x_1 is now truncated which cause the Kanban system to have lower average WIP level than Base Stock policy having equivalent parameter setting.

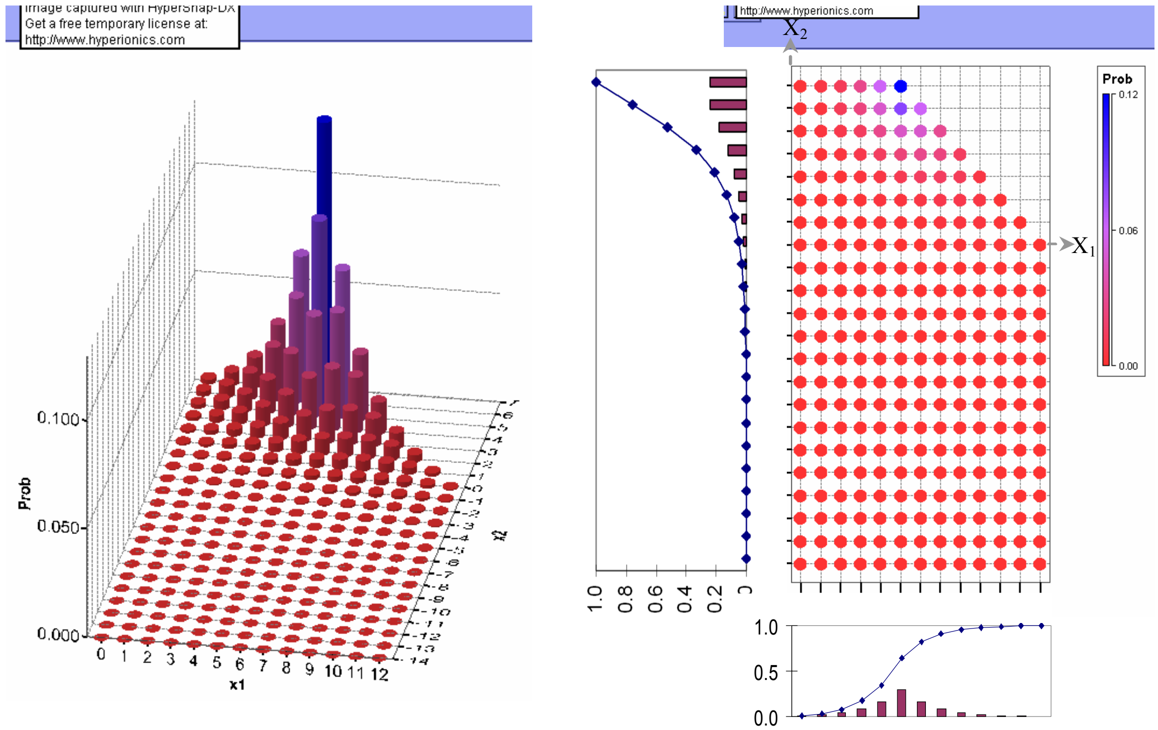
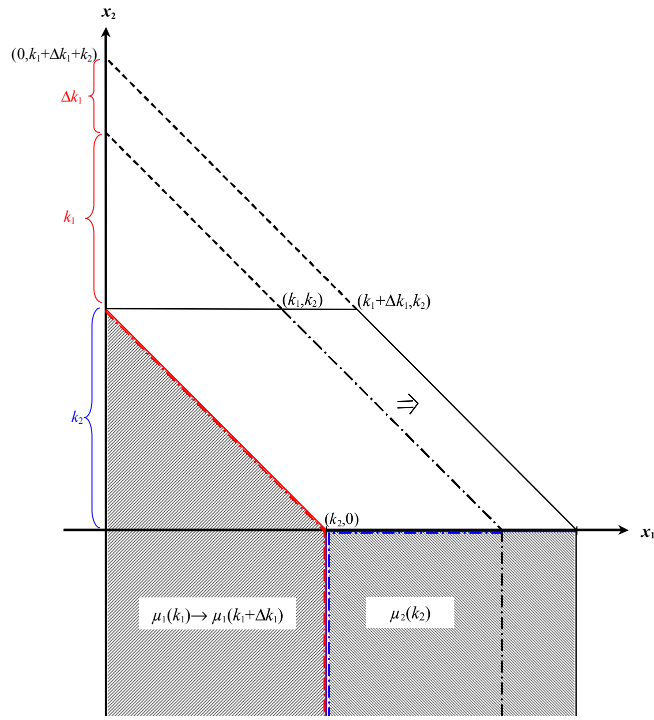
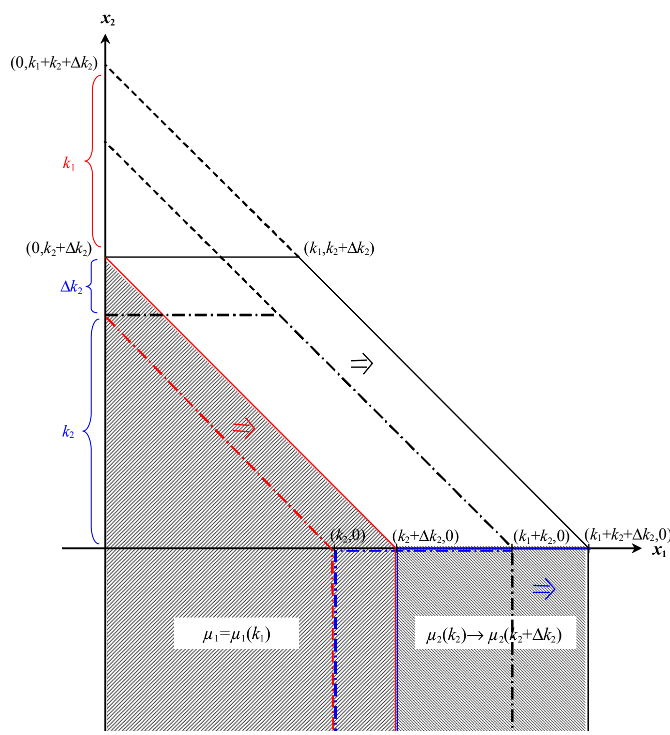


Figure 5-25: Probability density of two-stage Kanban system (k_1 = 6 and k_2 = 7)

Figure 5-26 shows the changes in the state space transition rate diagram when we changed k_1 to $k_1 + \Delta k_1$ (Figure 5-26a) and k_2 to $k_2 + \Delta k_2$ (Figure 5-26b). The dashed-dotted black line represents the old boundaries of the state space. The dashed-dotted red and blue lines represent the old transition rate boundaries of stage 1 and 2. The shaded region to the left of the red boundaries represents the states which the production rates of stage 1 reach its maximum at $\mu_1(k_1)$. Similarly, the shaded region to the right of the blue boundaries represents the states which the production rate of stage 2 reach its maximum at $\mu_2(k_2)$.



(a)



(b)

Figure 5-26: State-space of two-stage Base Stock system when parameters change

For changes in k_1, as k_1 increases, the probability that stage 2 will starve of input parts from stage 1 decreases. The service level of the system will increase and converge to the service level of a production system containing only stage 2 with k_2 amount of kanbans and infinite raw materials. However, as k_1 increase, the hedging point of the system also moves to the right causing the average WIP level to increase without any bound. As a result, the controller of the production line should not adjust k_1 as a primary way to adjust the service level; rather he should this adjustment as a fine tuning after a reasonable k_2 has been established.

For changes in k_2, as k_2 increases, the probability that stage 2 will be blocked because of not having enough stage-2 kanban decreases. Therefore, the amount of work-in-process in stage 2, y_2, depends only on the amount of k_1. If k_1 is large or approaching infinite, the amount of WIP at stage 2 converge to the amount of WIP of a similar production system operating under Base Stock policy where the basestock level of stage 2 is equal to k_2.

5.7.3 Control function of CONWIP-Kanban policy

CONWIP-Kanban combines the control mechanisms of Kanban and CONWIP together. It requires one parameter per stage and one parameter for the entire production line. Figure 5-27 shows the queueing network model of a two-stage four-machine CONWIP Kanban control system. There are C - k_1 amount of parts with both stage-1 Kanbans and CONWIP cards attached to them in buffer $\mathcal{B}$. Common practice is that the amount of CONWIP cards is less than the sum of the Kanban cards, thus, there will be k_1

$+ k_2 - C$ free-stage 1 kanbans in queue $\mathcal{K}_1$. See section 2.4 for detail dynamics of the system.

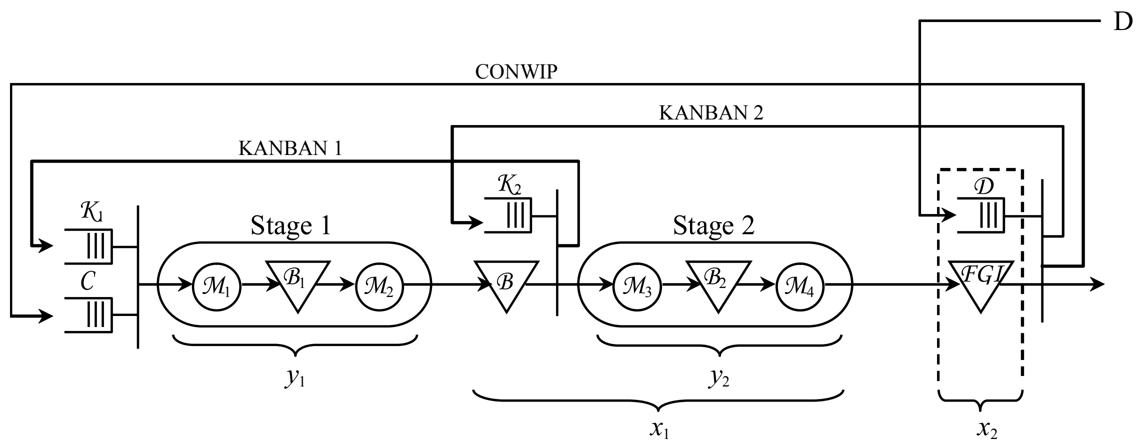


Figure 5-27: Two-stage CK control for two-stage four-machine production system

The limit boundaries and the production rates of each stage can be expressed as follow (see Appendix A for details).

$$x_1 + x_2 \le C \tag{5.32}$$

$$0 \le x_1 \le C \tag{5.33}$$

$$x_2 \le k_2 \tag{5.34}$$

$$\mu_1(y_1) \quad = \quad \begin{cases} \mu_1[C-(x_1+x_2)] & ; x_1+x_2 \ge C-k_1,\ x_2 \ge 0 \\ \mu_1(k_1) & ; x_1+x_2 < C-k_1,\ x_2 \ge 0 \end{cases} \tag{5.35}$$

$$\mu_1(y_1) \quad = \quad \begin{cases} \mu_1(C-x_1) & ; x_1 \ge C-k_1,\ x_2 \ge 0 \\ \mu_1(k_1) & ; x_1 < C-k_1,\ x_2 \ge 0 \end{cases} \tag{5.36}$$

$$\mu_2(y_2) \quad = \quad \begin{cases} \mu_2(k_2-x_2) & ; x_1+x_2 \ge k_2,\ x_2 \ge 0 \\ \mu_2(x_1) & ; x_1+x_2 < k_2,\ x_2 \ge 0 \end{cases} \tag{5.37}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ; x_1 \geq k_2,\, x_2 < 0 \\ \mu_2(x_1) & ; x_1 < k_2,\, x_2 < 0 \end{cases} \tag{5.38}$$

Using the transition boundaries and production rate approximations defined above, the state space transition diagram is shown in Figure 5-28.

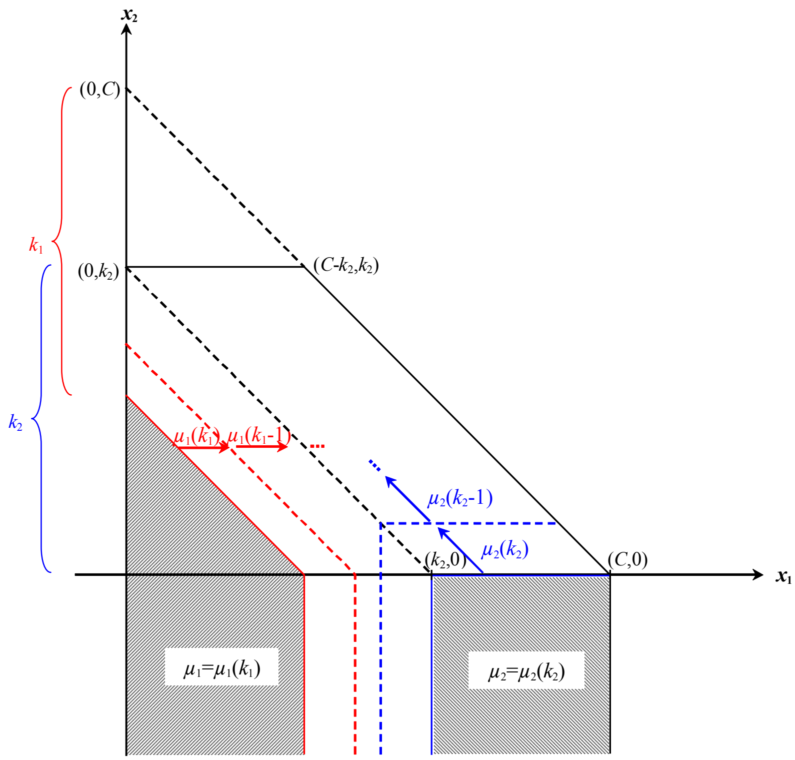


Figure 5-28: State-space transition rate diagram of CONWIP-Kanban system

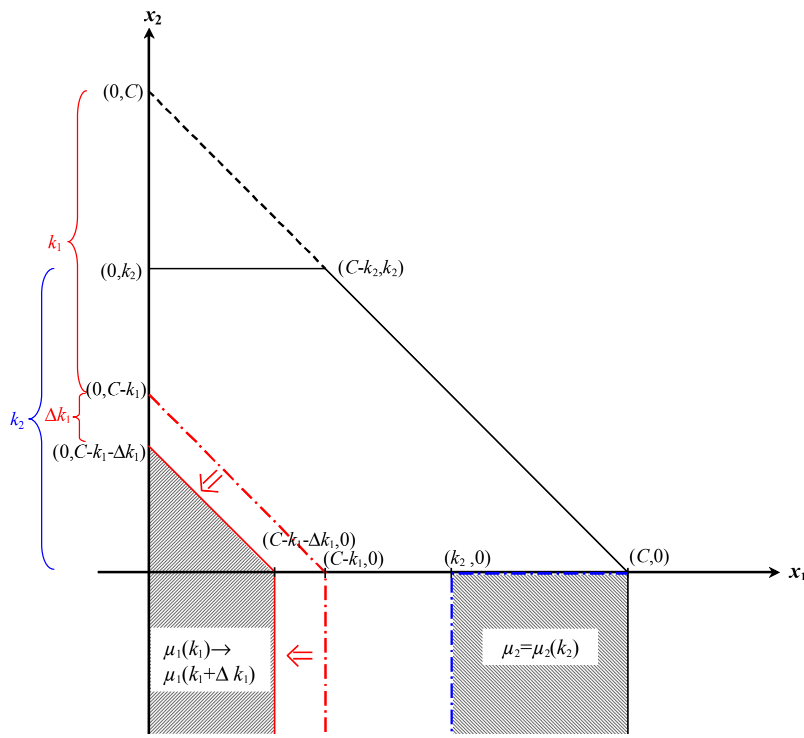
The shape of the state space of the CONWIP Kanban policy is similar to the state space of the Kanban policy. Therefore, the state space of CONWIP-Kanban system can be adjust to be similar to those in Figure 5-6 (a) and (d). This is because the switching curve of stage 1 has to change its slope at the x_1-axis as in the Kanban system. In a special case where $k_1 + k_2 = C$, they are identical.

However, there is a major difference in transition rate boundary. The CONWIP Kanban policy permit stage-1 transition rate boundary, the red solid line, to be independent of the amount of k_2, see Equation 5.35, 5.36. This flexibility permits the probability density of each state to be different from those of the Kanban policy.

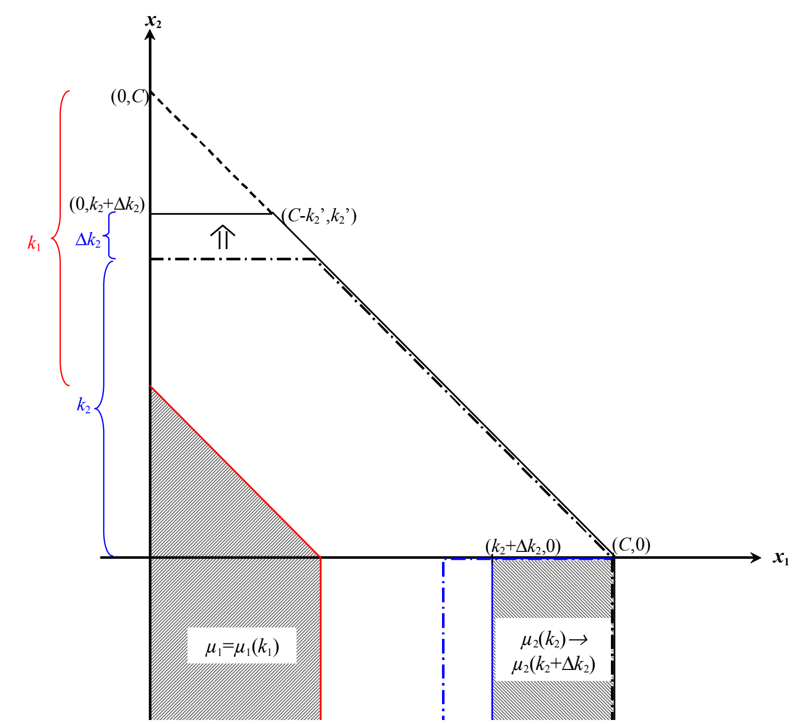
The probability density of each state in the CONWI-Kanban will look similar to those in the Kanban system, i.e. has peak density at (k_1, k_2), and then gradually reduces outward.

Figure 5-28 shows the changes in the state space transition rate diagram when we change k_1 to $k_1 + \Delta k_1$ (Figure 5-28a), k_2 to $k_2 + \Delta k_2$ (Figure 5-28b)

For changes in k_1, assuming that the system has the same amount of k_2 and C, the minimal amount of k_1 is one which is determined by the mechanism of Kanban control while the reasonable maximum amount of k_1 is equal to the amount of CONWIP cards, C. Note that any amount of stage-1 Kanban greater than the amount of CONIWP will give no difference in the performance of the system because the maximum amount of work-in-process is not only bounded by the amount of stage-1 Kanban but also bounded by the amount of CONWIP and the excess stage-1 kanban will be stored in queue $\mathcal{K}_1$ and never be used.



(a)



(b)

Figure 5-29: State-space of CONWIP Kanban when k_1 and k_2 change

As long as the amount of k_2 and C are held constant and $C \leq k_1 + k_2$, the shape of the state space of the system remains the same. However, as we look in the details of the transition rates of each state, the transition rate boundary of stage one, the red solid line, is move toward the left as k_1 increases and the maximum transition rate is now $\mu_1(k_1+\Delta k_1) \geq \mu_1(k_1)$. Therefore, as k_1 increases, the probability distribution is pushed toward the right resulting in higher average amount of work-in-process of the system and the probability that stage 2 will starve of input parts from stage 1 decreases; hence the service level increases.

For changes in k_2, assuming that the amount of k_1 and C are held constant, as k_2 increases, the shape of the state space of the system expands up causing the average finished goods inventory to increase. The hedging point of the system also moves to the left causing the amount of WIP to decrease. As the value of k_2 approaches the value of C, the shape of the state space of CONWIP-Kanban system will approach the shape of the state space of CONWIP system. The only difference is on the transition rate boundary of stage 1 except when k_1 is also approaching C. If $k_1 = k_2 = C$, the CONWIP-Kanban policy will be exactly the same as the CONWIP policy.

Figure 5-30 shows the changes in the state space transition rate diagram when we change C to $C + \Delta C$. As C increases, the state space expands to the right and stage-1 transition rate boundary is also shift to the right. This will result in higher service level because of larger amount of state above the x_1-axis, larger amount of work-in-process in the system, x_1, and a slightly higher amount of finished goods inventory. Note that if $C \geq k_1 + k_2$, the CONWIP control mechanism will play no role in the system and the state space transition rate diagram of the system will exactly be the same as the two-stage Kanban case.

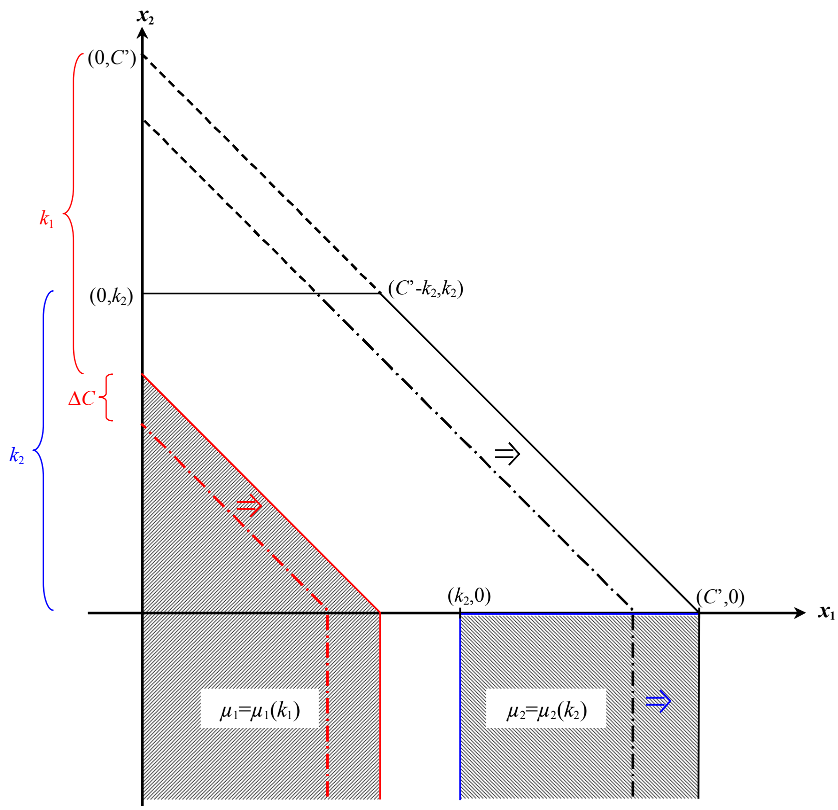


Figure 5-30: State-space of CONIWP Kanban when C changes

5.7.4 Control function of Extended Kanban policy

The Extended Kanban control is a combination of the Kanban and Base Stock control systems and includes each system as a special case. The system depends on two parameters per stage which are the number of kanbans and the base stock of finished parts in each stage. Figure 5-31 shows the queueing network model of a two-stage Extended Kanban control system. In this system, each stage 1 and 2 has k_1 and k_2 amount of kanbans that limit the maximum amount of parts in each stage. Buffer $\mathcal{B}$ and $\mathcal{FGI}$ has the target basestock level of s_1 and s_2 respectively.

The initial state of the system is having s_1 amount of work-in-process with stage-1 kanbans attached to them in buffer $\mathcal{B}$, and s_2 amount of finished goods with stage-2 kanbans attached to them in buffer $\mathcal{FGI}$. When a customer demand arrives, it is immediately splitted and transmitted to queue $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}$. See section 2.5 for detail dynamics of the system.

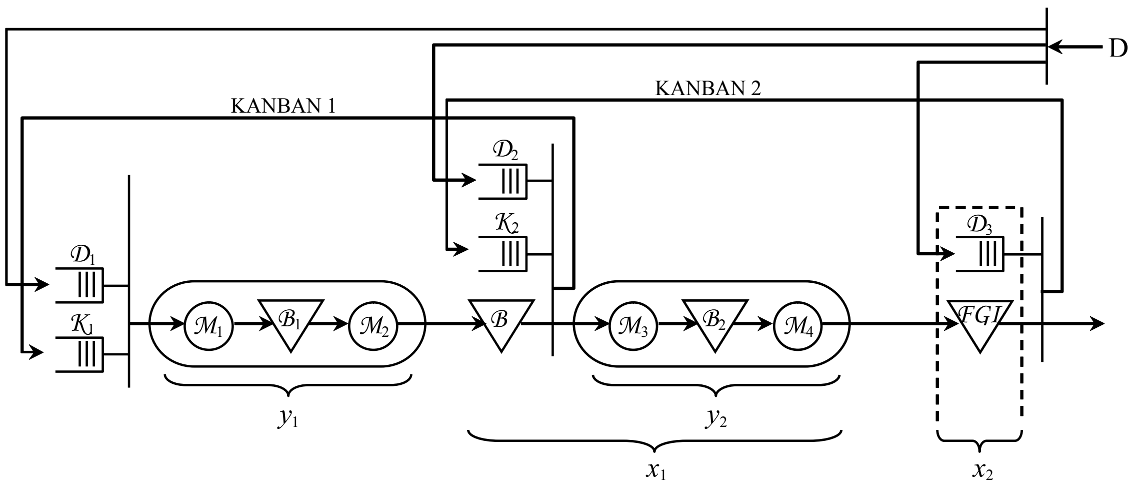


Figure 5-31: Two-stage EK control for two-stage four-machine production system

The limit boundaries and the production rates of each stage can be expressed as follow (see Appendix A for details).

$$x_1 + x_2 \le s_1 + s_2 \tag{5.39}$$

$$0 \le x_1 \le k_1 + k_2 \tag{5.40}$$

$$x_2 \le s_2 \tag{5.41}$$

$$\mu_1(y_1) = \begin{cases} \mu_1[k_1 + k_2 - x_1] & ; x_1 \ge k_2,\, x_2 < (s_1 + s_2) - (k_1 + k_2) \\ \mu_1[k_1] & ; x_1 < k_2,\, x_2 < (s_1 + s_2) - (k_1 + k_2) \end{cases} \tag{5.42}$$

$$\mu_1(y_1) = \begin{cases} \mu_1[(s_1+s_2)-(x_1+x_2)] & ;(x_1+x_2) \geq s_1+s_2-k_1 \\ \mu_1[k_1] & ;(x_1+x_2) < s_1+s_2-k_1 \end{cases} \tag{5.43}$$

$$\mu_2(y_2) = \begin{cases} \mu_2[s_2-x_2] & ;x_1+x_2 \geq s_2, x_2 \geq s_2-k_2 \\ \mu_2[x_1] & ;x_1+x_2 < s_2, x_2 \geq s_2-k_2 \end{cases} \tag{5.44}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ;x_1 \geq k_2, x_2 < s_2-k_2 \\ \mu_2(x_1) & ;x_1 < k_2, x_2 < s_2-k_2 \end{cases} \tag{5.45}$$

Using transition boundaries and production rates defined above, Figure 5-32 shows the state space transition rate diagram of the two-stage Extended Kanban system.

The shape of the state space of the Extended Kanban policy is similar to the shape of the state space of the Kanban policy; however, there are many differences in the positions of the boundaries of the state space and also the boundaries of the transition rate of each stage. First, the service level boundary is now defined by parameters s_1 and s_2 in stead of k_1 and k_2 as in the Kanban policy. Second, while the switching curve of the state space boundary of the Kanban policy changes its slope from -1 to ∞ when $x_2 = 0$ and $x_1 = k_1 + k_2$, the Extended Kanban mechanisms permits changing the slope when $x_2 < 0$ and $x_1 = k_1 + k_2$. This implies that the Inventory limit and the buffer limit can now be specified independently. Third, the transition rate boundary of stage 1, the red solid line, starts at the point $(0, s_1 + s_2 - k_1)$ and changes it slope from -1 to ∞ when $x_1 = k_2$ and $x_2 = (s_1 + s_2) - (k_1 + k_2)$. Last, the transition rate boundary of stage 2, the blue solid line, starts at the point $(s_1 + k_2, s_2 - k_2)$ and changes it slope from 0 to ∞ when $x_1 = k_2$ and $x_2 = s_2 - k_2$.

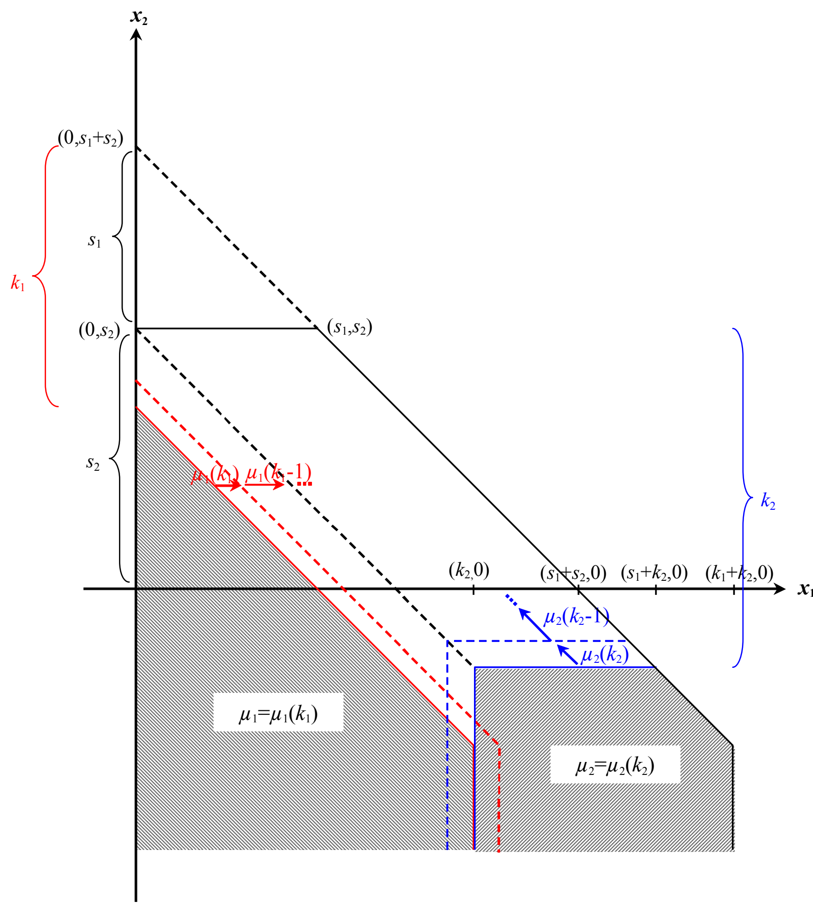
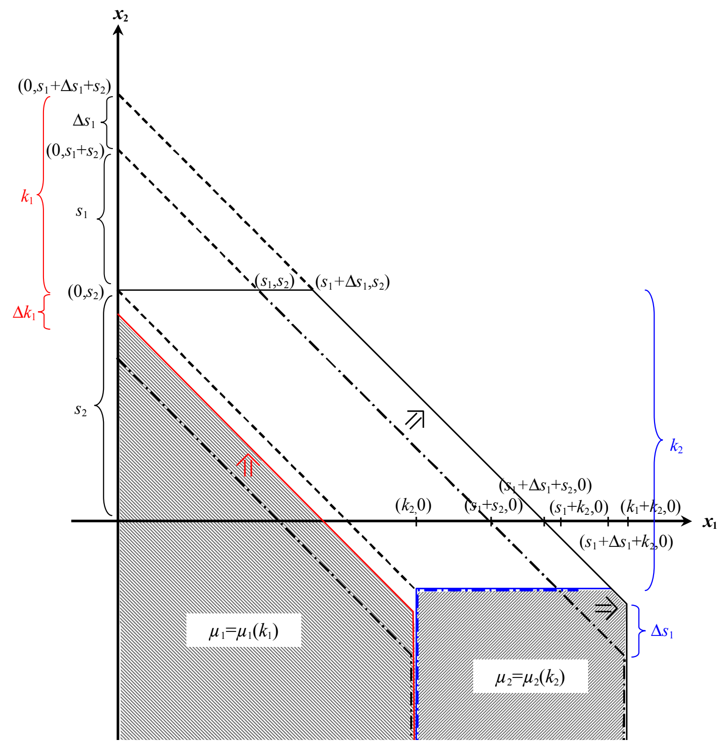


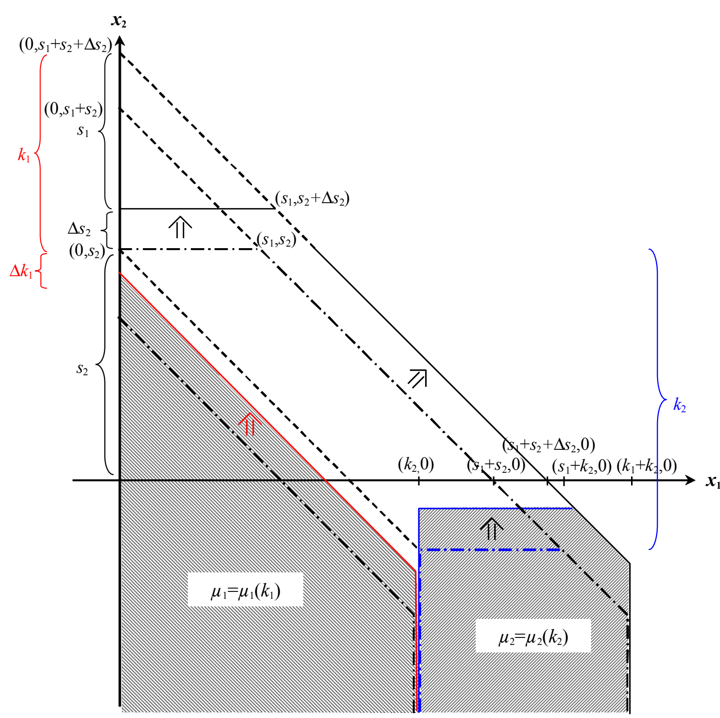
Figure 5-32: State-space transition rate diagram for Extended Kanban system

By comparing the state space of Extended-Kanban system with the optimum, it can be adjust to be similar to those in Figure 5-6 (a), (c), (d), and (f). This is because the switching curve of stage 1 is allowed to change its slope at the x_1-axis or below the x_1-axis.

Figure 5-33 shows the changes in the state space transition rate diagram when we change s_1 to $s_1 + \Delta s_1$ (Figure 5-33a), and s_2 to $s_2 + \Delta s_2$ (Figure 5-33b).



(a)



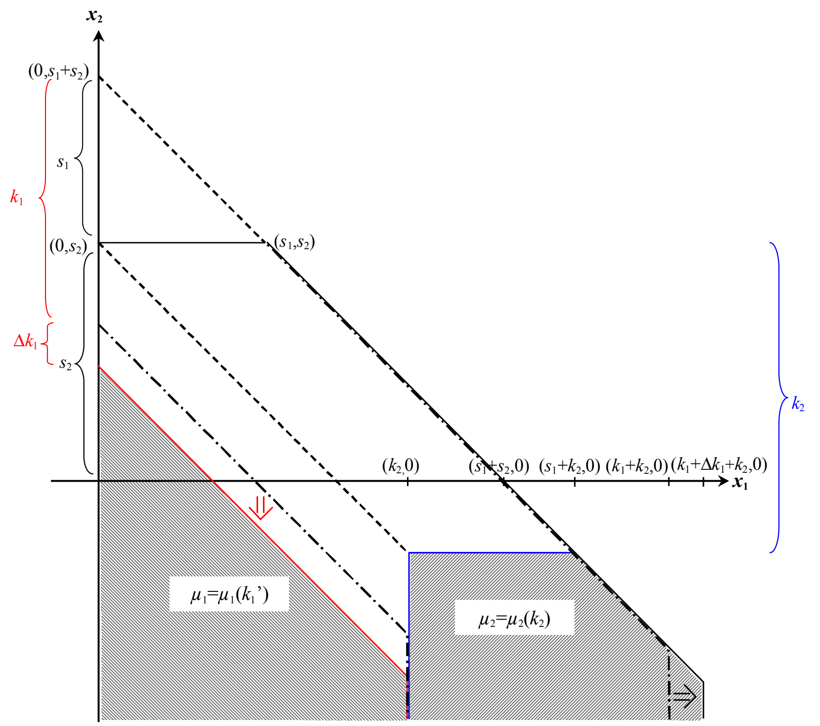
(b)

Figure 5-33: State space of Extended Kanban when s_1 and s_2 change

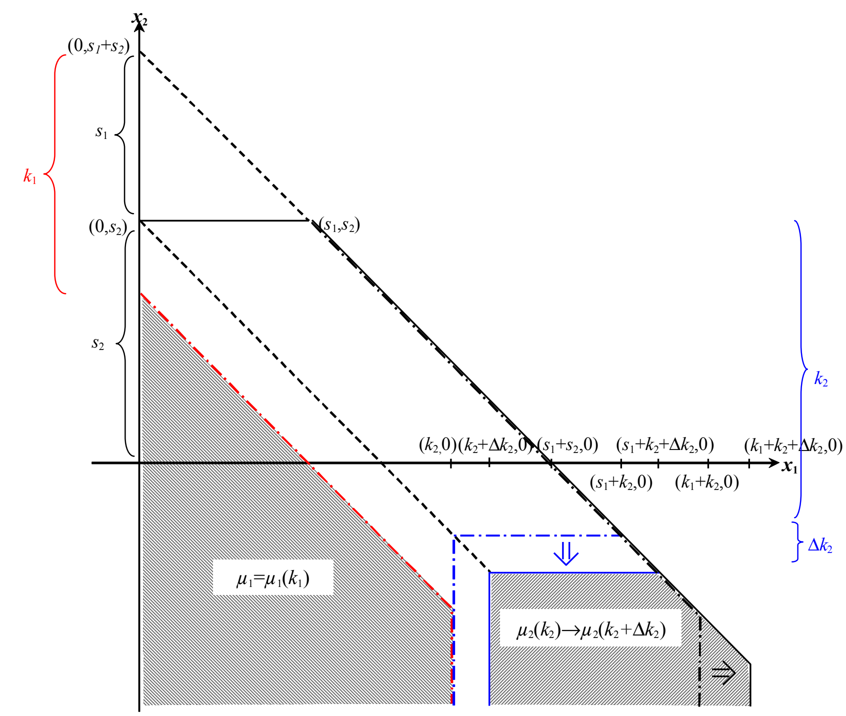
For changes in s_1, assuming that other parameters are held constant, the minimal amount of s_1 is zero and the maximum amount is k_1 according to the mechanism of the Extended Kanban policy. Any amount of s_1 greater than k_1 is invalid because every part holds in buffer $\mathcal{B}$ has to have one stage-1 Kanban attached to it. As s_1 increases, the number of states expands to the right and the starting point of the transition rate boundary of stage 1 shifts upward; hence, the probability distribution of the state space is pushed toward the right resulting in higher average amount of work-in-process of the system, x_1. Also, as s_1 increases, the probability that stage 2 will starve of input parts from stage 1 decreases and hence the service level increases.

For the basestock level of the second stage, s_2, the minimal amount of s_2 is again equal to zero and the maximum amount is k_2. Similar reasoning for these boundaries as in s_1 applies here. As s_2 increases, the number of states above the x_1-axis expands both vertically and horizontally. Also, both of the transition rate boundaries of the two stages shift upward. This can be viewed as shifting the x_1-axis downward from the original position in Figure 5-33 without changing any probability density of each state. As a result, the average finished goods inventory, $E[x_2^+]$, increases by the same amount as the increase in s_2 while the amount of work-in-process, $E[x_1]$, remains the same.

Figure 5-34 shows the changes in the state space transition rate diagram when we change k_1 to $k_1 + \Delta k_1$ (Figure 5-34a), and k_2 to $k_2 + \Delta k_2$ (Figure 5-34b).



(a)



(b)

Figure 5-34: State space of Extended Kanban when k_1 and k_2 change

For changes in k_1, the minimal amount of k_1 is one which is determined by the mechanism of Kanban control while the maximum amount of k_1 is not limited. As k_1 increases, see Figure 5-34a, the number of states above the x_1-axis remains the same, while the total number of states increases and expands to the right. The stage-1 transition rate boundary shifts downward with higher limiting rate, while the transition rate boundary of stage 2 is at the same position. As a result, the probability distribution is pushed toward the right resulting in higher average amount of work-in-process of the system, x_1. Also, as k_1 increases, the probability that stage 2 will starve of input parts from stage 1 decreases and hence the service level increases. As k_1 keeps increasing, the service level of the system will converge to the service level of a production system containing only stage 2 with k_2 amount of kanbans and s_2 basestock level with infinite raw materials. Similar analysis in finding the service level as k_1 increases to infinite in the Kanban policy can be applied here.

For changes in k_2, the minimal amount of k_2 is equal to one which is determined by the mechanism of Kanban control while the maximum amount of k_2 is not limited. As k_2 increases, see Figure 5-34b, the number of states above the x_1-axis remains the same, while the total number of states increases and expands to the right as in the case of changing k_1. The stage-1 transition rate boundary is now changing its slope from -1 to ∞ at the new point, i.e. the boundary expands to the right. The transition rate boundary of stage 2 shifts 45 degree downward the right with higher limiting transition rate. As a result, average amount of work-in-process, x_1, increases.

Figure 5-35 shows simulation results of two-stage Extended Kanban system varying stage-1 kanban from 4 to 6, stage-2 kanban from 10 to 12, stage-1 basestock level from 2 to 3 and stage-2 basestock level of 6 and 8. Blue dots represent the amount of finished goods inventory and red dots represent the amount of work-in-process. This verifies the explanations of effect of changing the amount of kanbans and basestock levels in previous section. The amount of finished goods inventory and service level depend heavily on the amount of stage-2 basestock level. However, the amount of work-in-process is not affected by stage-2 basestock level The amount of finished goods inventory and service level also increase when the amount of stage-1 basestock level increases but not as much as the case of varying amount of stage-2 kanbans.

One can view that the amount of kanbans is like a fine tuning on the performance of the system to achieve minimal cost for a given service level constraint.

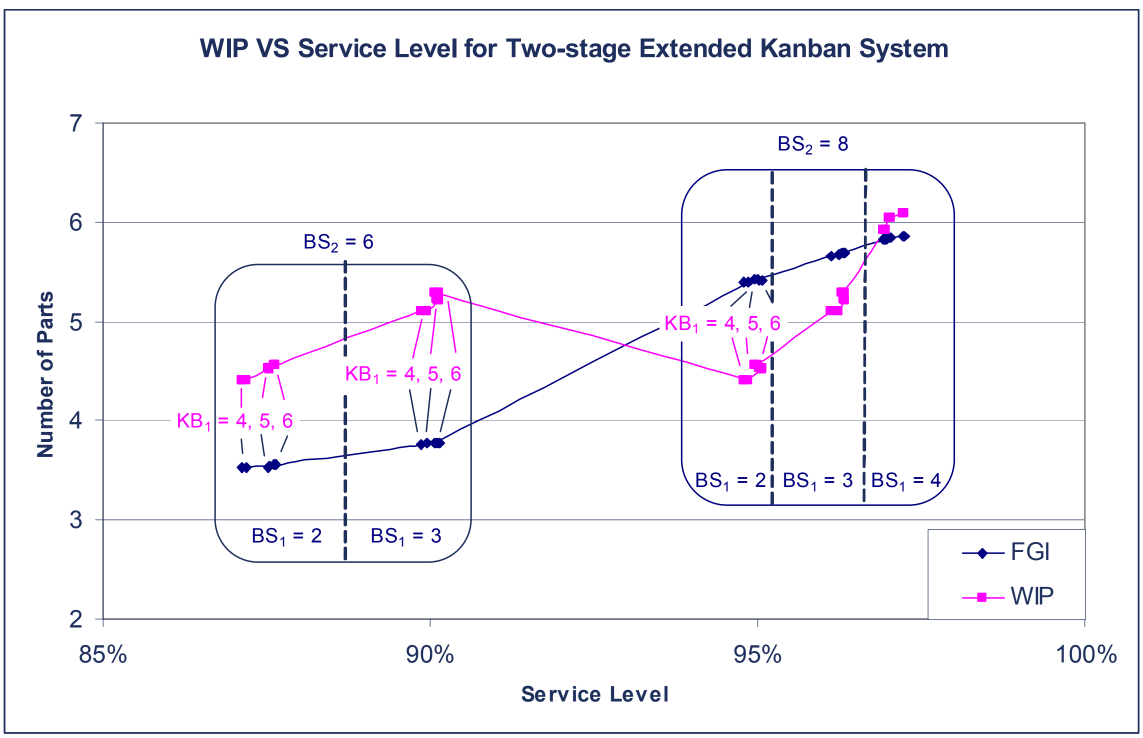


Figure 5-35: WIP vs service level for two-stage Extended Kanban system

5.7.5 Control function of Generalized Kanban policy

Generalized Kanban policy requires two control parameters per stage, which are the amount of kanbans and the target basestock level of each stage as same as in the Extended Kanban control. It also includes the classical Kanban and Base Stock control systems as a special case. Figure 5-36 shows the queueing network model of a two-stage four-machine Generalized Kanban control system.

The difference between this Generalized Kanban control and the Extended Kanban control is that the amount of kanban of each stage is independent with the amount of base stock of that stage, i.e. the amount of kanban has no need to be greater than the target basestock level as in the Extended Kanban control.

The initial state of the system is the same as in the Extended Kanban system, i.e. having s_1 amount of work-in-process with stage-1 kanbans attached to them in buffer $\mathcal{B}$, and s_2 amount of finished goods with stage-2 kanbans attached to them in buffer $\mathcal{FGI}$. When a demand arrives, it will be splitted and transmitted to queue $\mathcal{D}_2$ and $\mathcal{D}$. Detail dynamics of the system can be found in section 2.6.

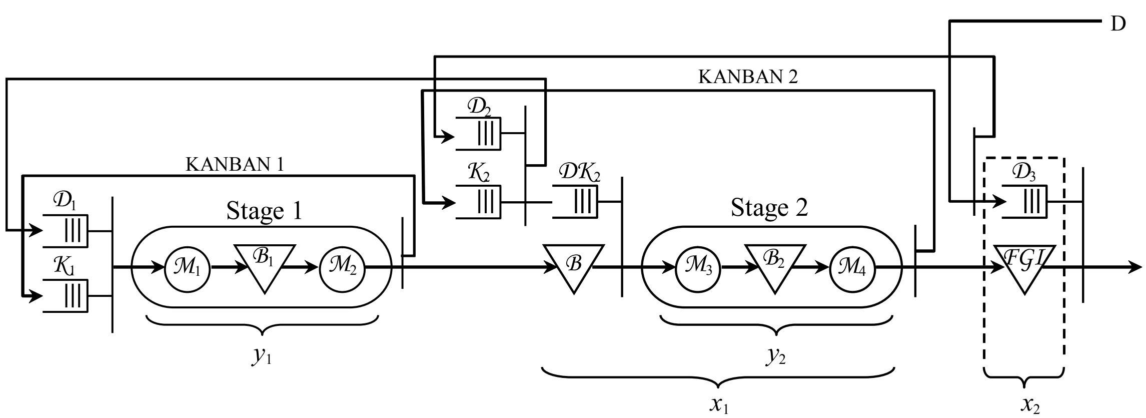


Figure 5-36: Two-stage GK control for two-stage four-machine production system

The limit boundaries and the production rates of each stage can be expressed as follow (see Appendix A for details).

$$x_1 + x_2 \le s_1 + s_2 \tag{5.46}$$

$$0 \le x_1 \le s_1 + k_2 \tag{5.47}$$

$$x_2 \le s_2 \tag{5.48}$$

$$\mu_1(y_1) \quad = \quad \begin{cases} \mu_1[(s_1 + s_2) - (x_1 + x_2)] & ;(x_1 + x_2) \ge s_1 + s_2 - k_1 \\ \mu_1[k_1] & ;(x_1 + x_2) < s_1 + s_2 - k_1 \end{cases} \tag{5.49}$$

$$\mu_1(y_1) \quad = \quad \begin{cases} \mu_1(s_1 + k_2 - x_1) & ;x_1 \ge s_1 + k_2 - k_1 \\ \mu_1(k_1) & ;x_1 < s_1 + k_2 - k_1 \end{cases} \tag{5.50}$$

$$\mu_2(y_2) \quad = \quad \begin{cases} \mu_2[s_2 - x_2] & ;x_1 + x_2 \ge s_2, x_2 \ge s_2 - k_2 \\ \mu_2[x_1] & ;x_1 + x_2 < s_2, x_2 \ge s_2 - k_2 \end{cases} \tag{5.51}$$

$$\mu_2(y_2) \quad = \quad \begin{cases} \mu_2(k_2) & ;x_1 \ge k_2, x_2 < s_2 - k_2 \\ \mu_2(x_1) & ;x_1 < k_2, x_2 < s_2 - k_2 \end{cases} \tag{5.52}$$

Figure 5-37 shows the state space transition rate of the two-stage Generalized Kanban system for the case where $s_1 \le k_1$ and $s_2 \le k_2$. The shape of this state space is similar to the shape of the state spaces of the Kanban and Extended Kanban policies. However, there are many similarities and differences in the boundaries of the state space and also the boundaries of the transition rate of each stage compared with those two policies.

First, the service level boundaries are defined by s_1 and s_2 as in the Extended Kanban policy in stead of k_1 and k_2 as in the Kanban policy.

Second, the switching curve of the state space changes its slope from -1 to ∞ when $x_1 = s_1 + k_2$ and $x_2 = s_2 - k_2$. Unlike the Extended Kanban policy which restricts the slope to change only when $x_2 \le 0$, this policy allows the curve to change it slope above the x_1-axis or below the x_1-axis. Therefore, the shape of the Generalized Kanban can be adjusted to be the same as those in Figure 5-6 (a) – (f).

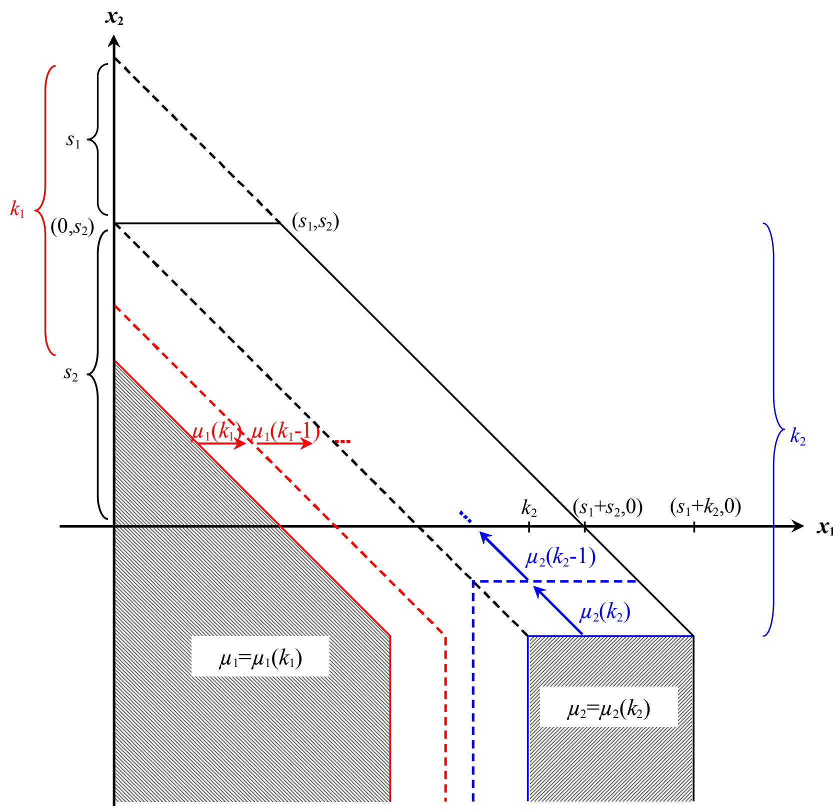


Figure 5-37: State-space transition rate diagram of Generalized Kanban system

Third, the transition rate boundary of stage 1, the red solid line, starts at the point $(0, s_1 + s_2 - k_1)$ as in the Extended Kanban policy; however, it changes its slope from -1 to ∞ when $x_1 = s_1 + k_2 - k_1$ and $x_2 = s_2 - k_2$. This implies that the position of the switching point of the transition rate boundary does not only depend on k_2 as in the Extended Kanban policy, but also depend on s_1 as well.

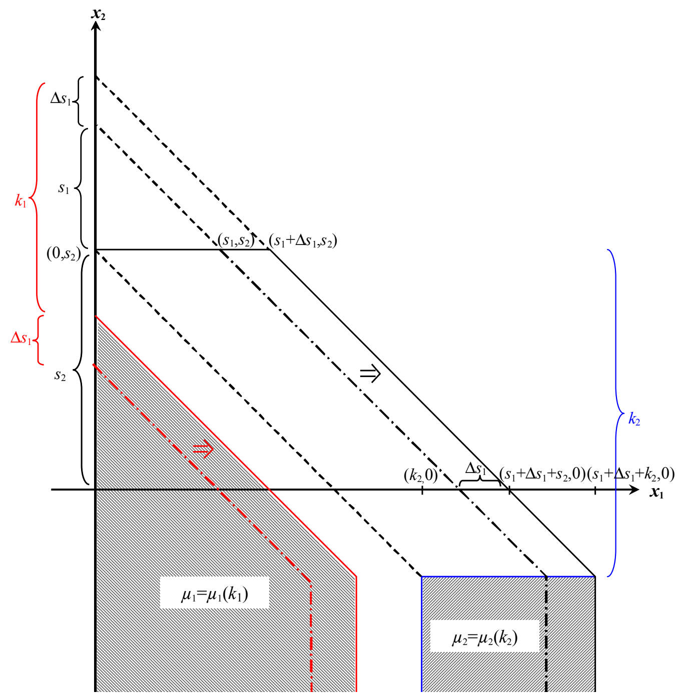
Last, the transition rate boundary of stage 2, the blue solid line, starts at the point $(s_1 + k_2, s_2 - k_2)$ and changes it slope from 0 to ∞ when $x_1 = k_2$ and $x_2 = s_2 - k_2$ which is the same as in the Extended Kanban policy.

Figure 5-38 shows the changes in the state space transition rate diagram when we change s_1 to $s_1 + \Delta s_1$ (Figure 5-38a), and s_2 to $s_2 + \Delta s_2$ (Figure 5-38b).

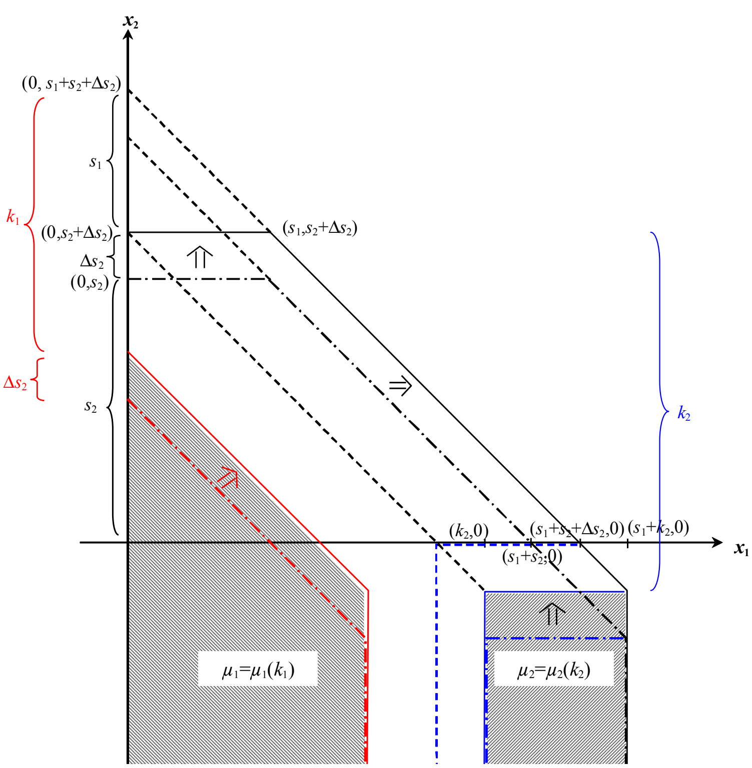
For changes in s_1, assuming that other parameters are held constant, as s_1 increases, the number of states expands to the right and the starting point of the transition rate boundary of stage 1 shifts upward; hence, the probability distribution of the state space is pushed toward the right resulting in higher average amount of work-in-process of the system, x_1. Also, as s_1 increases, the probability that stage 2 will starve of input parts from stage 1 decreases and hence the service level increases.

For the basestock level of the second stage, s_2, as s_2 increases, the number of states above the x_1-axis expands both vertically and horizontally. Also, both of the transition rate boundaries of the two stages shift upward. This can be viewed as shifting the x_1-axis downward from the original position in Figure 5-38 without changing any probability density of each state. As a result, the average finished goods inventory increases by the same amount as the increase in s_2 while the amount of work-in-process, remains the same.

Figure 5-39 shows the changes in the state space transition rate diagram when we change k_1 to $k_1 + \Delta k_1$ (Figure 5-39a), and k_2 to $k_2 + \Delta k_2$ (Figure 5-39b).

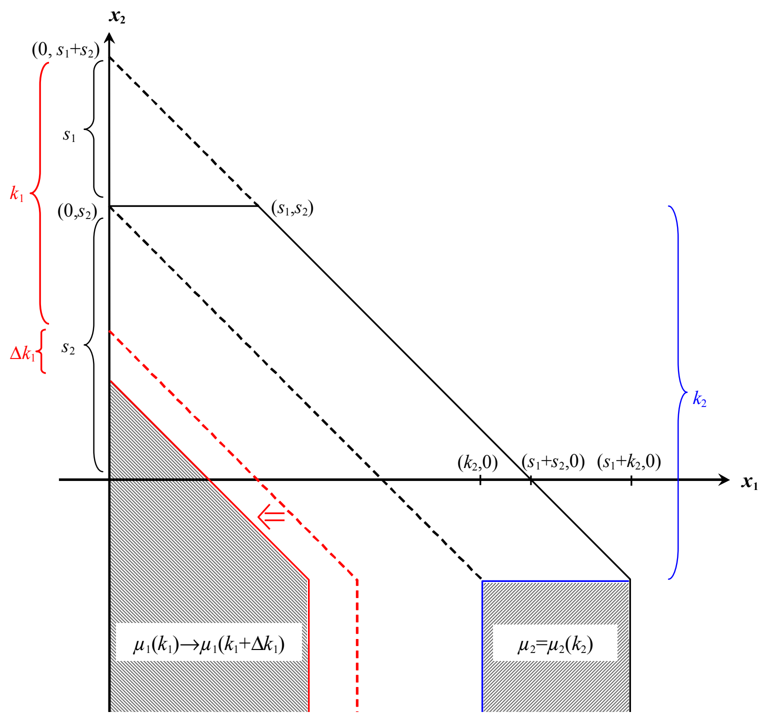


(a)

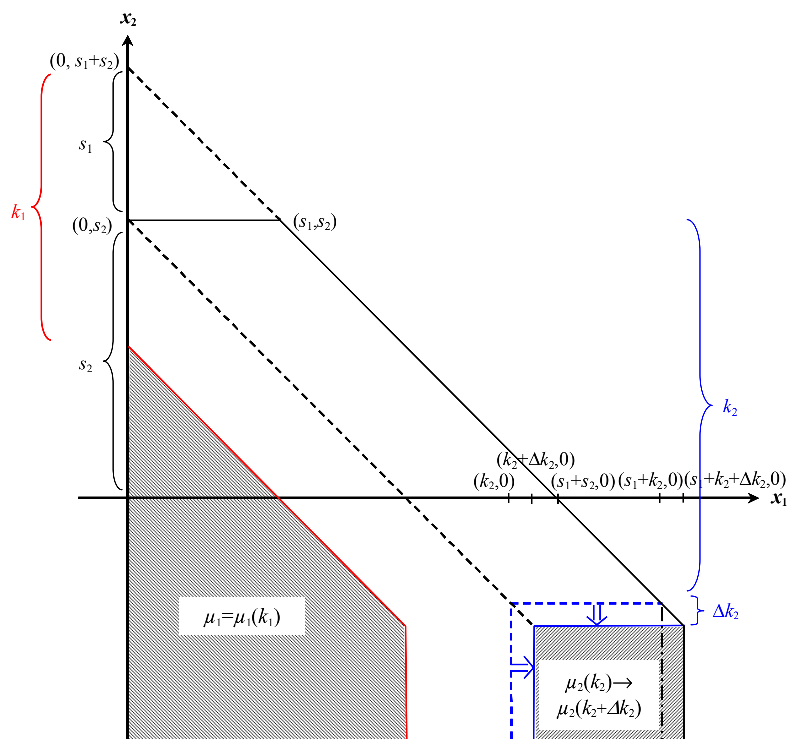


(b)

Figure 5-38: State-space of Generalized Kanban when s_1 and s_2 change



(a)



(b)

Figure 5-39: State-space of Generalized Kanban when k_1 and k_2 change

For changes in k_1, the minimal amount of k_1 is one which is determined by the mechanism of Kanban control while the maximum amount of k_1 is not limited. As k_1 increases, the number of states above the x_1-axis and the total number of states remains the same. The stage-1 transition rate boundary shifts downward with higher limiting rate, while the transition rate boundary of stage 2 is at the same position. As a result, the probability distribution is pushed toward the right resulting in higher average amount of work-in-process of the system, x_1. Also, as k_1 increases, the probability that stage 2 will starve of input parts from stage 1 decreases and hence the service level increases. As k_1 keeps increasing, the service level of the system will converge to the service level of a production system containing only stage 2 with k_2 amount of kanbans and s_2 basestock level with infinite raw materials. Similar analysis in finding the service level as k_1 increases to infinite in the Kanban policy can be applied here.

For changes in k_2, the minimal amount of k_2 is equal to one which is determined by the mechanism of Kanban control while the maximum amount of k_2 is not limited. As k_2 increases, the number of states above the x_1-axis remains the same, while the total number of states increases and expands to the right as in the case of changing k_1. The stage-1 transition rate boundary is now changing its slope from -1 to ∞ at the new point $(s_1 + k_2' - k_1, s_2 - k_2')$, i.e. the boundary expands to the right. The transition rate boundary of stage 2 shifts 45 degree downward the right with higher limiting transition rate. As a result, average amount of work-in-process, x_1, increases.

Figure 5-40 shows simulation results of two-stage Generalized Kanban system varying stage-1 kanban from 4 to 6, stage-2 kanban from 10 to 12, stage-1 basestock level from 2 to 3 and stage-2 basestock level of 6 and 8. Blue dots represent the amount of finished goods inventory and red dots represent the amount of work-in-process. This is the same set of parameter as in Extended Kanban case. This verifies the explanations of effect of changing the amount of kanbans and basestock levels in previous section. The effects of each parameter change are quite similar to Extended Kanban case except having slightly different actual value. Again, one can view that the amount of kanbans is like a fine tuning on the performance of the system to achieve minimal cost for a given service level constraint.

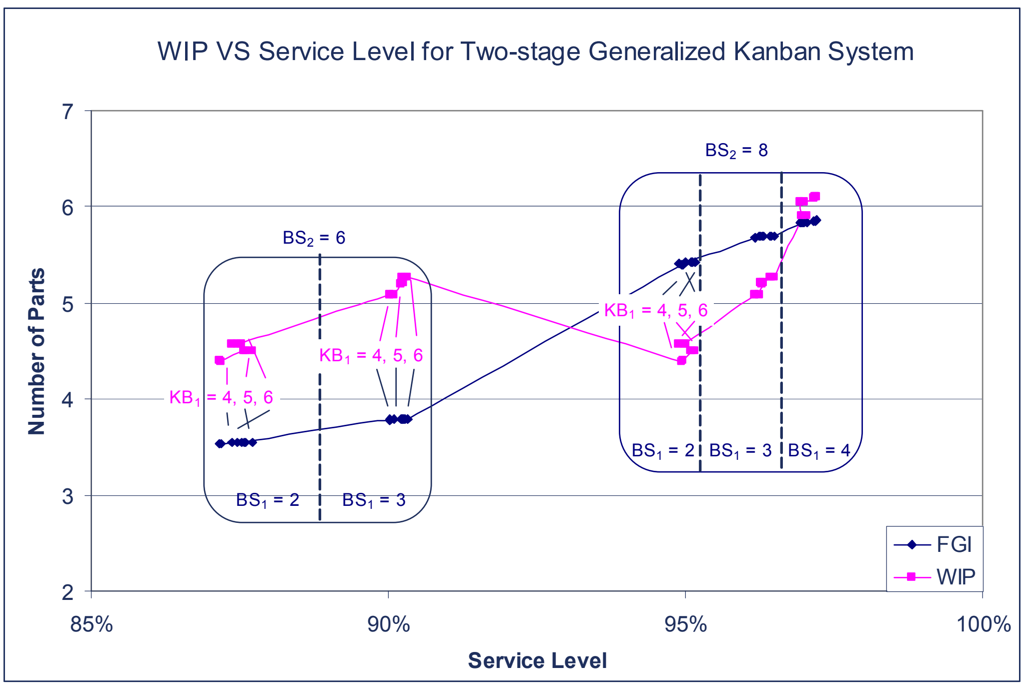


Figure 5-40: WIP vs service level for two-stage Generalized Kanban system

5.7.6 Control function of Extended CONWIP Kanban policy

Extended CONWIP Kanban policy combines Base Stock, CONWIP, and Kanban mechanism together. It requires two parameters per stage with one additional parameter for the whole production line. Figure 5-42 shows the queueing network model of a two-stage four-machine Extended CONWIP Kanban control system.

In this system, stage one and two has k_1 and k_2 amount of kanbans that limit the maximum amount of parts in each stage. Buffer $\mathcal{B}$ and $\mathcal{FGI}$ has the target basestock level of s_1 and s_2 respectively. The CONWIP limits the maximum amount of work-in-process of the entire production stage.

When a demand arrives, it will be splitted and transmitted to queue $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ immediately. Detail dynamics of the system and its properties can be found in chapter 3.

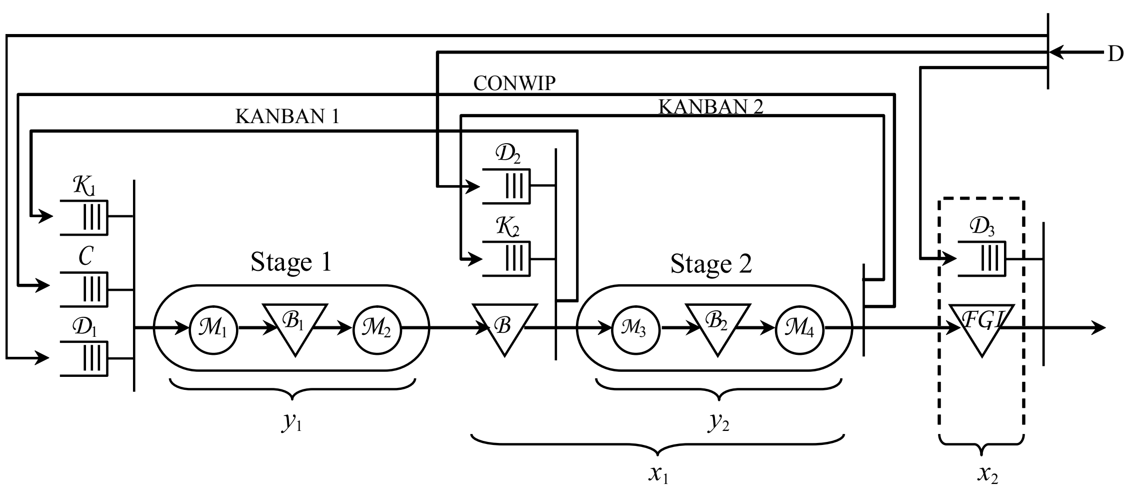


Figure 5-41: Two-stage ECK control for two-stage four-machine production system

The limit boundaries and the production rates of each stage can be expressed as follow (see Appendix A for details).

$$x_1 + x_2 \le s_1 + s_2 \tag{5.53}$$

$$0 \le x_1 \le C \tag{5.54}$$

$$x_2 \le s_2 \tag{5.55}$$

$$\mu_1(y_1) = \begin{cases} \mu_1(C - x_1) & ; x_1 \ge C - k_1 \\ \mu_1(k_1) & ; x_1 < C - k_1 \end{cases} \tag{5.56}$$

$$\mu_1(y_1) = \begin{cases} \mu_1[(s_1 + s_2) - (x_1 + x_2)] & ; x_1 + x_2 \ge s_1 + s_2 - k_1 \\ \mu_1[k_1] & ; x_1 + x_2 < s_1 + s_2 - k_1 \end{cases} \tag{5.57}$$

$$\mu_2(y_2) = \begin{cases} \mu_2[s_2 - x_2] & ; x_1 + x_2 \ge s_2, x_2 \ge s_2 - k_2 \\ \mu_2[x_1] & ; x_1 + x_2 < s_2, x_2 \ge s_2 - k_2 \end{cases} \tag{5.58}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ; x_1 \ge k_2, x_2 < s_2 - k_2 \\ \mu_2(x_1) & ; x_1 < k_2, x_2 < s_2 - k_2 \end{cases} \tag{5.59}$$

Figure 5-42 shows the state space and transition rate diagram of the two-stage Extended CONWIP Kanban system when $s_1 + k_2 = C$. The shape of this state space is similar to the shape of the state spaces of the Generalized Kanban policy. The followings are the key characteristic of the ECK policy.

For the transition boundaries, the finished goods limit is defined by s_2. The Inventory limit is defined by s_1 and s_2. Hence, it can be independently adjusted from the finished goods limit. Finally, the buffer limit is defined by C and therefore can be independently adjusted from the finished goods and Inventory limit. As a result, by adjusting parameters s_1, s_2, and C, the shape of the state space can be identical to the approximated optimum two-boundary control policy in Figure 5-6 (a) – (f).

In addition, the maximum production rates of stage 1 and stage 2 are limited by k_1 and k_2 which are independent from the parameters defining the transition boundary.

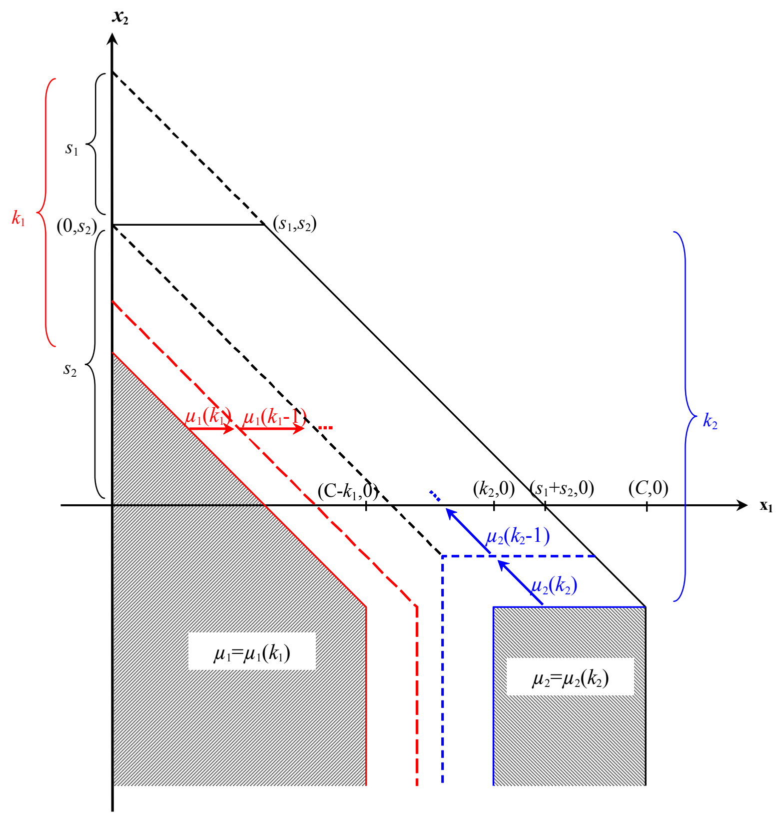
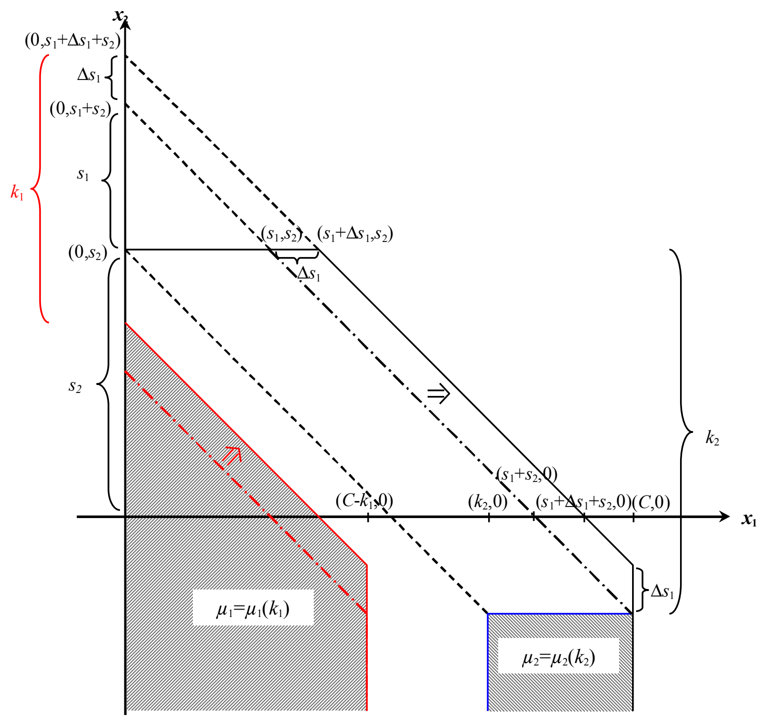


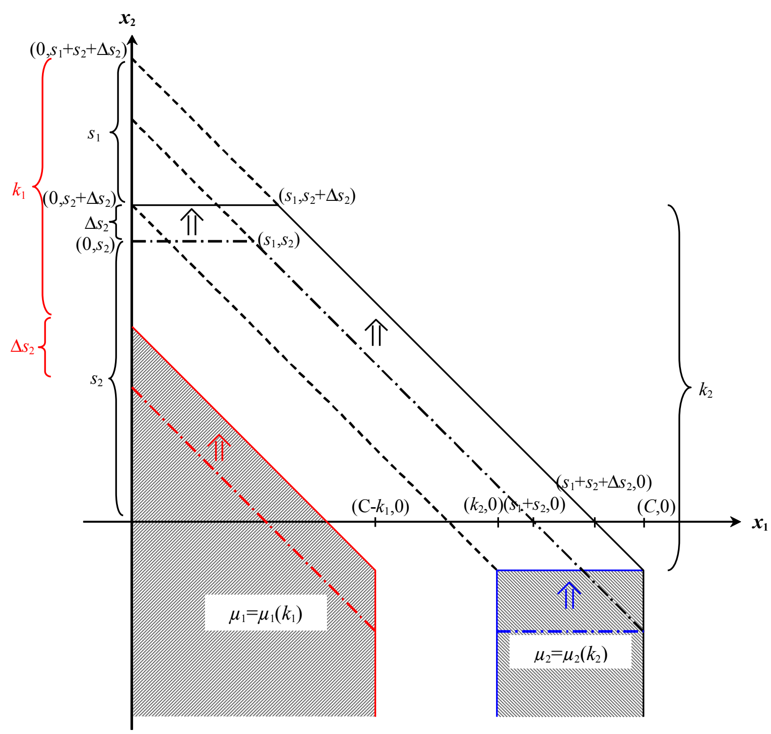
Figure 5-42: State-space transition rate diagram of ECK system

Figure 5-43 shows the changes in the state space transition rate diagram when we change s_1 to $s_1 + \Delta s_1$ (Figure 5-43a), and s_2 to $s_2 + \Delta s_2$ (Figure 5-43b).

For changes in s_1, as s_1 increases, the number of states, the transition rate boundary of stage 1, and the hedging point move to the right; hence, the probability distribution of the state space is pushed toward the right resulting in higher average amount of work-in-process of the system, x_1. Also, as s_1 increases, the probability that stage 2 will starve of input parts from stage 1 decreases and hence the service level increases.



(a)



(b)

Figure 5-43: State-space of ECK system when s_1 and s_2 change

For the basestock level of the second stage, s_2, as s_2 increases, the number of states above the x_1-axis expands both vertically and horizontally. Also, both of the transition rate boundaries of the two stages shift upward. This can be viewed as shifting the x_1-axis downward from the original position in Figure 5-42 without changing any probability density of each state. As a result, the average finished goods inventory, increases by the same amount as the increase in s_2 while the amount of work-in-process remains the same.

Figure 5-44 shows the changes in the state space transition rate diagram when we change k_1 to $k_1 + \Delta k_1$ (Figure 5-44a), and k_2 to $k_2 + \Delta k_2$ (Figure 5-44b).

As k_1 increases, the number of states above the x_1-axis and the total number of states remains the same. The stage-1 transition rate boundary shifts downward with higher limiting rate, while the transition rate boundary of stage 2 is at the same position. As a result, the probability distribution is pushed toward the right resulting in higher average amount of work-in-process of the system, x_1. Also, as k_1 increases, the probability that stage 2 will starve of input parts from stage 1 decreases and hence the service level increases.

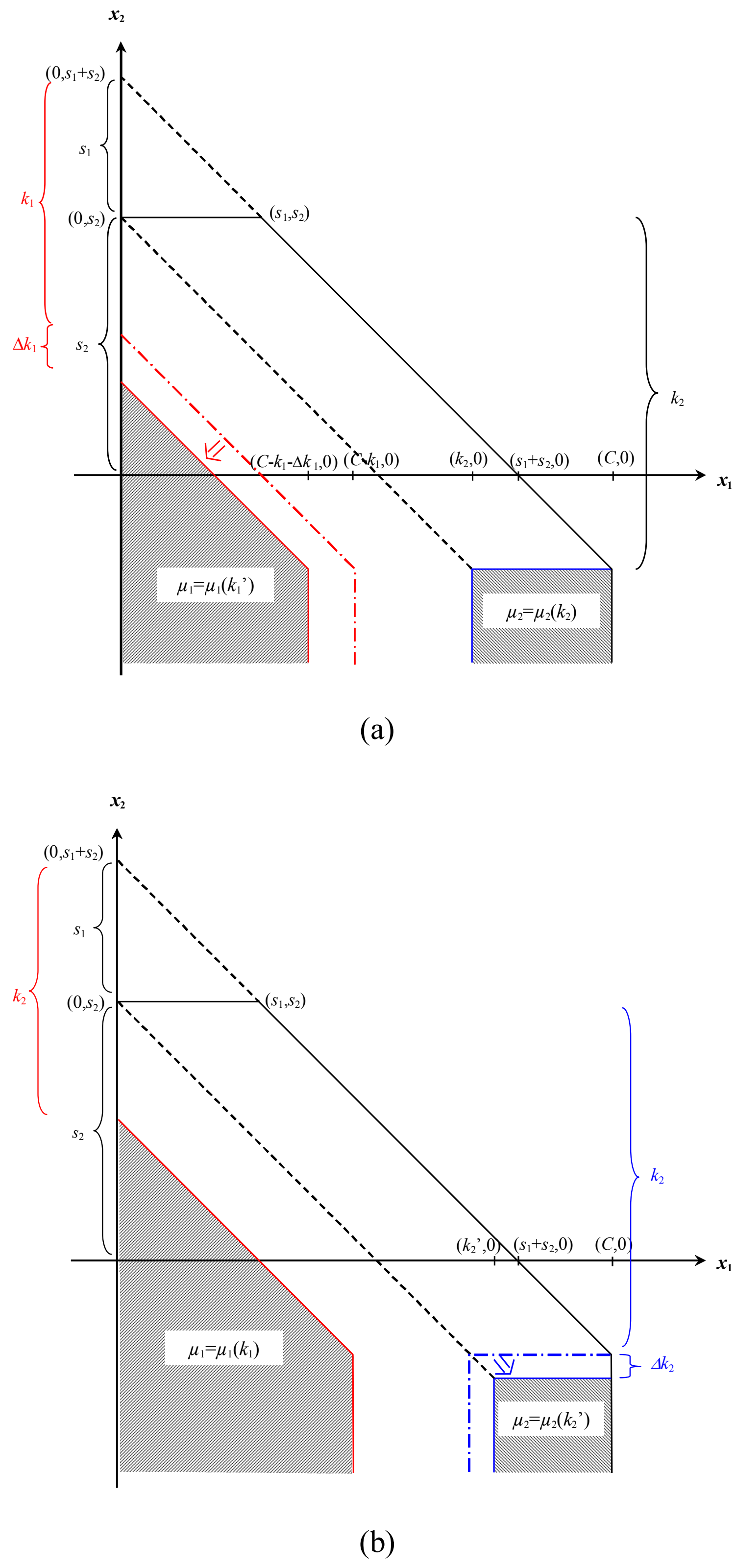


Figure 5-44: State-space of ECK system when k_1 and k_2 change

For changes in k_2, as k_2 increases the total number of states remains the same, while the transition rate boundary of stage 2 shifts 45 degree downward the right with higher limiting transition rate. As a result, average amount of finished goods inventory increases while the amount of work-in-process decreases.

Figure 5-45 shows the changes in the state space transition rate diagram when we change C to $C + \Delta C$. As C increases, the state space, stage-1 transition rate and stage-2 transition rate expand to the right.

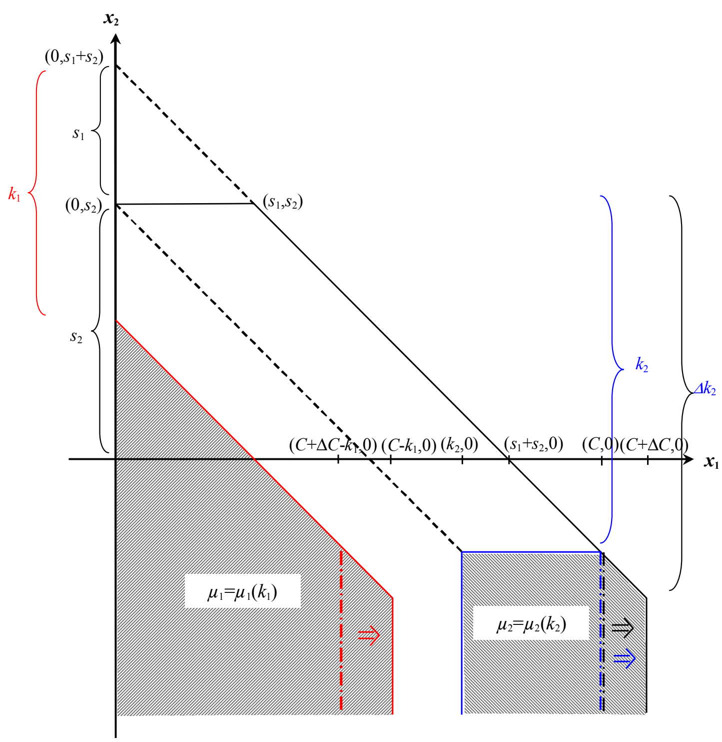


Figure 5-45: State-space of ECK system when C changes

5.8 Discussion on two-stage control policies

In Table 5.2, we summarized the effect of control parameters of two-stage control policies on the transition boundaries and the production rate. Sources of equations are also noted under each parameter. It can be seen that for all control policies considered, the Inventory limit can be adjusted separately once the finished goods limit has been defined. For example, one can define the finished goods limit of the Kanban policy using parameter k_2, then adjusts the Inventory limit through parameter k_1. As a result, the hedging point of all policies considered here can be adjusted and has not necessary to be on the x_2-axis as in the single-stage control policies.

For Kanban and CONWIP Kanban policies, the Inventory limit and buffer limit are also defined by the same parameter. This causes the turning point of the switching curve of stage 1 to be on the x_1-axis. On the other hand, for the Extended Kanban, Generalized Kanban and Extended CONWIP Kanban, the buffer limit can be adjusted after the Inventory limit has already been defined.

Table 5.2: Comparison of two-stage control policies

Control Policies	Transition Boundaries			Production rate	
	FGI limit	WIP limit	Buffer limit	Stage 1	Stage 2
Single-stage control policies					
CONWIP	C (5.8)	C (5.6)	C (5.7)	C (5.9)	C (5.10)
Single-stage Base Stock	s (5.13)	s (5.12)	-	-	-
Single-stage ECK	s (5.18)	s (5.16)	C (5.17)	s, C (5.19)	-
Two-stage control policies					
Base Stock	s_2 (5.22)	s_1, s_2 (5.21)	-	s_1, s_2 (5.23)	s_2 (5.24)
Kanban	k_2 (5.27)	k_1, k_2 (5.25)	k_1, k_2 (5.26)	k_1, k_2 (5.28 - 5.29)	k_2 (5.30 - 5.31)
CONWIP Kanban	k_2 (5.34)	C (5.32)	C (5.33)	k_1, C (5.35 - 5.36)	k_2 (5.37 - 5.38)
Extended Kanban	s_2 (5.41)	s_1, s_2 (5.39)	k_1, k_2 (5.40)	s_1, s_2, k_1, k_2 (5.42 - 5.43)	s_2, k_2 (5.44 - 5.45)
Generalized Kanban	s_2 (5.48)	s_1, s_2 (5.46)	s_1, k_2 (5.47)	s_1, s_2, k_1, k_2 (5.49 - 5.50)	s_2, k_2 (5.51 - 5.52)
Extended CONWIP Kanban	s_2 (5.55)	s_1, s_2 (5.53)	C (5.54)	s_1, s_2, k_1, C (5.56 - 5.57)	s_2, k_2 (5.58 - 5.59)

For example, one can define the finished goods limit of the Generalized Kanban policy using parameter s_2 and define the Inventory limit through parameter s_1, then adjusts the buffer limit through parameter k_1. As a result, the turning point of the switching curve of stage-1 in these policies can be adjusted to be either above, below or on the x_1-axis Note that for the Extended Kanban, the turning point is required to be either below or on the x_1-axis due to its mechanism.

Lastly, the production rate of stage 1 in Extended Kanban and Generalized Kanban can also be adjusted separately from the transition boundaries, while the production rate of stage 2 can not. In the Extended CONWIP Kanban policy, the production rate of both stage 1 and 2 can be adjusted separately from the transition boundaries. Therefore, the Extended CONWIP Kanban policy has the most flexibility on the system configuration. Not only it can adjust all of the transition boundaries separately, but it can also adjust the production rate of each stage separately.

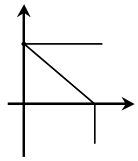
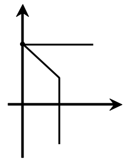
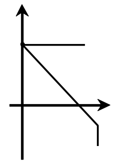
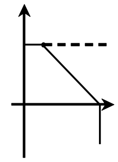
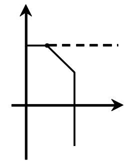
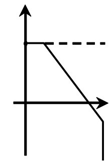
In Table 5.3, we summarized the parameter dependency of each control policy in an easier format. Reading from the left, the check marks represent the ability to independently adjust that column from the columns to the left of it. For example, in CONWIP Kanban policy, once the finished goods limit has been defined, the Inventory limit and the production rate of stage 1 can be separately adjusted while the buffer limit and the production rate of stage 1 can not.

Table 5.3: Parameter dependency comparison of all control policies

Control Policies	Transition Boundaries			Production rate	
	FGI limit	WIP limit	Buffer limit	Stage 1	Stage 2
Single-stage control policies					
CONWIP	✓	×	×	×	×
Single-stage Base Stock	✓	×	×	×	×
Single-stage ECK	✓	×	✓	×	×
Two-stage control policies					
Base Stock	✓	✓	×	×	×
Kanban	✓	✓	×	×	×
CONWIP Kanban	✓	✓	×	✓	×
Extended Kanban	✓	✓	✓	✓	×
Generalized Kanban	✓	✓	✓	✓	×
Extended CONWIP Kanban	✓	✓	✓	✓	✓

In Table 5.4, we compare the ability of each control policy to adjust the shape of its state space with the shaped of the state space that can be found under the two-boundary control

Table 5.4: State-space comparison of all control policies

State-space comparison						
Single-stage control policies						
CONWIP	✓	×	×	×	×	×
Single-stage Base Stock	×	×	×	×	×	×
Single-stage ECK	✓	✓	✓	×	×	×
Two-stage control policies						
Base Stock	×	×	×	×	×	×
Kanban	✓	×	×	✓	×	×
CONWIP Kanban	✓	×	×	✓	✓	×
Extended Kanban	✓	✓	✓	✓	×	✓
Generalized Kanban	✓	✓	✓	✓	✓	✓
Extended CONWIP Kanban	✓	✓	✓	✓	✓	✓

The Extended CONWIP Kanban and the Generalized Kanban policies are the most adjustable policy. Note that since Base Stock policy does not have buffer limit, it can never be adjusted to the shape close to the optimal ones.

CHAPTER 6

ON THE DESIGN OF ECK CONTROL SYSTEMS

In Extended CONWIP Kanban control system; there are two design parameters per stage and one additional parameter for the entire production line. Thus, if the manufacturing system is composed of N stages, there are $2N + 1$ parameters that must be determined. An efficient control system must provide a good trade-off between the cost of shortage and carrying finished goods or WIP. The results in Chapter 5 suggested that the amount of WIP is mostly influenced by the basestock levels in each internal stage, the distribution of WIP is influenced by the amount of kanbans and CONWIPs, and shortage is influenced by the basestock level of the last stage and also the amount of CONWIPs.

The aim of this chapter is to find these parameters. We first investigate the influence of each design parameter in special cases of Extended CONWIP Kanban control policies. We applied analytical and/or simulation methods to show tradeoffs between basestock levels, numbers of kanbans, numbers of CONWIPs, production capacity, and customer service level. We then present design rules based on the insights from our investigations.

This chapter is organized as follows. Section 6.1 provides an analysis of single-stage ECK control system and its optimal design parameters. This is followed by analysis of multi-stage ECK control system in Section 6.2 which is based on analytical and experimental results of two-stage ECK control system. Section 6.3 is devoted to a special case of two-stage ECK control system when there is a bottleneck station. Finally, the conclusion on the effects of each parameter is provided in Section 6.4.

6.1 Single-stage ECK control system

In this section we consider a single-stage ECK control system discussed in section 5.5.3. There are only two design parameters to be specified: the amount of CONWIP card C, and the finished goods basestock level s. We first find the production capacity upper bound in a saturated ECK control system and investigate the effect of these two design parameters in a saturated ECK control system to find the maximum production capacity. We then analyzed the special case when C equals to s. This special case allows us to search for optimum solution. Finally, we relax the restriction to allow C and s to differ.

6.1.1 Single-stage saturated ECK control system

A saturated system is the system under the assumption of infinite number of raw parts and customer demands. It will provide the production capacity of the system.

Theorem 6.1 The production capacity of a single-stage ECK system is a function of the number of CONWIP C, but is independent of the target basestock level s.

Proof Since demand queues in the system are always non-negative; it can be eliminated from its synchronization station. Once the demand queues are eliminated, a saturated single-stage ECK system is a closed queueing having a population of C customers. Therefore, the production capacity of a single-stage ECK system only depends on the number of CONWIP C, not the basestock level s.

If processing times are exponential, this production capacity has a product form solution. Moreover, if all stations are identical, the production capacity μ_{max} is given by Hopp and Spearman (1996):

$$\mu_{\max} = \frac{C}{C + W_0 - 1} \cdot r_b \tag{6.1}$$

Where

r_b is bottleneck production rate,

W_0 is the critical WIP level and is equal to the number of stations, and

C is the number of CONWIPs, which equals to the numbers of parts in the system.

The production capacity of the system can be made as close as possible to the bottleneck rate r_b by increasing the number of CONWIP C. In order to achieve bottleneck production capacity, a large number of CONWIP is needed when $m > 1$.

6.1.2 Effect of C (or s) in a single-stage unsaturated ECK control system when $C = s$

When $C = s$, the system is equivalent to a single-stage CONWIP control system. If we assume that demands arrive according to a Poisson process with rate λ_D, the finished goods exit the manufacturing facility will follow a state-dependent Markovian process whose rate, $\lambda_P(n)$, depends on the number of parts in the manufacturing facility, n, as shown in Figure 6-1.

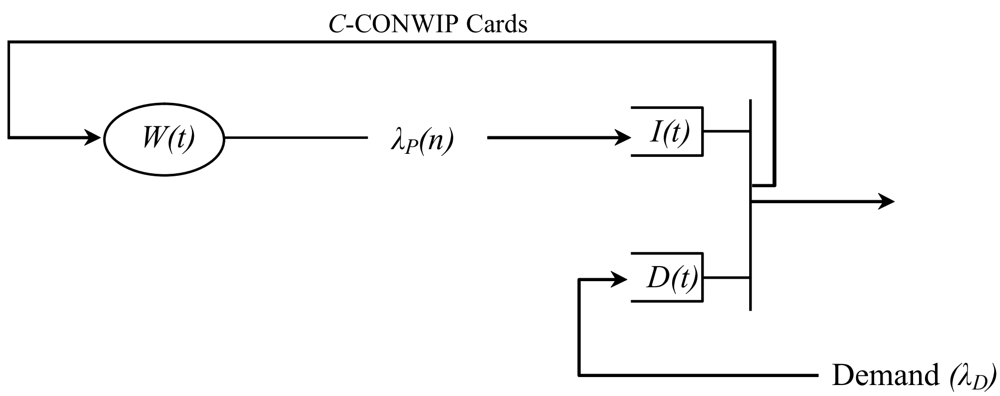


Figure 6-1: Single-stage CONWIP System.

Assumed that there is a holding cost of $\$h$ per unit per time unit for inventory held in the system, either in the form of work-in-process or in the form of finished good inventory and the customer service level is constrained to be higher than a specified level above α. Our optimization problem is now in the form of

$$Min\ Cost(C) \quad = \quad h \cdot C$$

$$\text{s.t} \quad P\{I_C > 0\} \geq \alpha \text{. Or } P\{D_C > 0\} \leq 1 - \alpha$$

Since the expected WIP level, $E[W_C(t)]$, and the expected finished goods level, $E[I_C(t)]$, are increasing in C and the expected backordered demand, $E[D_C(t)]$ is decreasing in C, the service level is increasing function in C, *i.e.* $P\{I_{C+1} > 0\} \geq P\{I_C > 0\}$ (Dallery and Liberopoulos 1995). Thus, there exists C^* such that Cost (C^*) < Cost (C^*+1) and the service level of (C^*+1) is less than the specified service level α. Then, the solution can be found by searching. We first need to find the initial values.

Dallery and Liberopoulus (1995) suggested an iterative algorithm to find C^*. The algorithm starts using $C^0 = C_{\min}$ where $\frac{\lambda_D}{\lambda_P(C_{\min})} < 1$. Then compute the underlying Markov chain at the synchronization stations to find the service level. C^* is the minimal CONWIP level that can obtain the specified service level constraint. To find a reasonable lower bound of $C_{\min}$, we can again use Equation 6.1. By setting $\mu_{\max} = \lambda_D$, the resulting WIP level is a lower bound of $C_{\min}$.

We propose a simpler way to find a reasonable value of C^0. Since we know that single-stage CONWIP system is equivalent to single-stage Base stock system with maximum-Inventory limit, the optimal base stock of single-stage Base stock system is therefore lower than or equal to the optimal CONWIP level in single-stage CONWIP system (Dallery and Liberopoulos 2002). Hence, we can set C^0 equal to the optimal base stock in single-stage Base stock system. The optimal basestock level, s^*, is given by the well-known critical ratio in the newsvendor problem, i.e. it is the smallest integer that satisfies (see Rubio and Wein 1996):

$$P(WIP \leq s^*) \geq \alpha \tag{6.2}$$

If the facility consists of a single-server station with exponential service rate μ, the optimal basestock level is given by (see Buzacott and Shanthikumar (1993), Section 4.3.1; Rubio and Wein (1996))

$$s^* = \frac{1}{\ln \rho}\left[\ln\left(\frac{h}{b+h}\right)\right] \tag{6.3}$$

Where h = holding cost per unit time

b = backordering cost per unit time

$\frac{h}{b+h}$ = 1 - α where α is the service level

In the case where the facility consists of a Jackson network of servers, s^* satisfies a non-closed-form expression that can be solved numerically.

After obtaining a good initial CONWIP level $C^0 = s^*$, we will increase the amount of CONWIPs by one unit, while observing the customer service level, until the observed service level is greater than the required service level. Because of the monotone property between the service level and the amount of CONWIP, the minimal CONWIP cards that satisfy the service level constraint will be optimum. The most computationally demanding task is to find or estimate the service level. This task can be accomplished using real-time data or can be done off line using an analytically tractable approximation or simulation.

6.1.3 Effect of C and s when $C \neq s$

When a replenishment production order is placed, it is not immediately authorized to be released in the facility until the WIP in the system is below a given WIP-cap of C parts. If the WIP in the system is at or above C, the order is put on hold until the WIP drops below C. Notice that this single-stage ECK control system is equivalent to the single-stage generalized kanban policy (Buzacott 1989; Zipkin 1989) and basestock with WIP cap policy (Liberopoulus and Dallery 2002). The optimal basestock level is the optimal basestock level found in a similar production system operating under Base stock policy and the optimal CONWIP level C is the minimal CONWIP level such that the customer service level is still satisfied while maintaining the same basestock level. Combining this property with the cost structure found in section 6.1.1, we propose the following searching scheme.

Algorithm 6.1:

Step 0: Set $s^0 = C^0 = C^*$

Step 1: Loop $s^0 = s^0 - 1$ until $P\{I_C > 0\} < \alpha$

Step 2: Loop $C^0 = C^0 + 1$ until $P\{I_C > 0\} < \alpha$

Step 3: Repeat Step 1 and 2 until the service level is unattainable at a particular basestock level regardless of the amount of CONWIPs.

Step 4: The final C^* and s^* are last values that the service level was attained

6.2 Effect of k_i, s_i, and C in a multi-stage ECK control system

An N-stage ECK control system has $2N + 1$ design parameters: k_i and s_i for $i = 1, \ldots, N$ and C. Let us use the vector notations: $\boldsymbol{K} = (k_1, k_2, \ldots, k_N)$, $\boldsymbol{S} = (s_1, s_2, \ldots, s_N)$ and $\boldsymbol{C} = (C)$. We would like to fine the effect of these design parameters on the main performance measures, especially the production capacity and the proportion of backordered demands. Obviously, this analysis is more difficult than in the single-stage case. We begin by illustrating the effect of $\boldsymbol{K}$, $\boldsymbol{S}$, and $\boldsymbol{C}$ by means of an example. We then prove properties for the general case.

6.2.1 Effect of k_i and C with exponential processing time

Consider an ECK control system composed of N identical stages in tandem. The service times of all the machines are exponentially distributed with identical service rate μ equal to 1. We find the production capacity, from the saturated system, and the proportion of backordered demands, by analyzing the last synchronization station.

Production capacity

Figure 6-2 shows the saturated queueing network model of the saturated version of the ECK control system having N stages in series from the original queueing model adopted from Figure 3-1, shown in Figure 6-2. We removed the following queues in the adaptation. Queue $\mathcal{P}_0$ in contains an infinite amount of raw parts and queue $\mathcal{D}_i$, $i = 1,\ldots, N + 1$, have an infinite number of demands. Therefore, queue $\mathcal{P}_0$ and, $\mathcal{D}_i$, $i = 1,\ldots, N + 1$ play no role in the synchronization station they belong to, since they never block the transfer of parts through those synchronization station; hence they can be eliminated. Once queue $\mathcal{D}_{N+1}$ has been eliminated from the synchronization station, $J_{N,N+1}$, queue $\mathcal{BK}_N$ remains the only queue in $J_{N,N+1}$. If there is only one queue feeding the synchronization station, this queue can be eliminated, since any customer or part entering this queue will immediately go through the synchronization station. The resulting queueing network is shown in Figure 6-2.

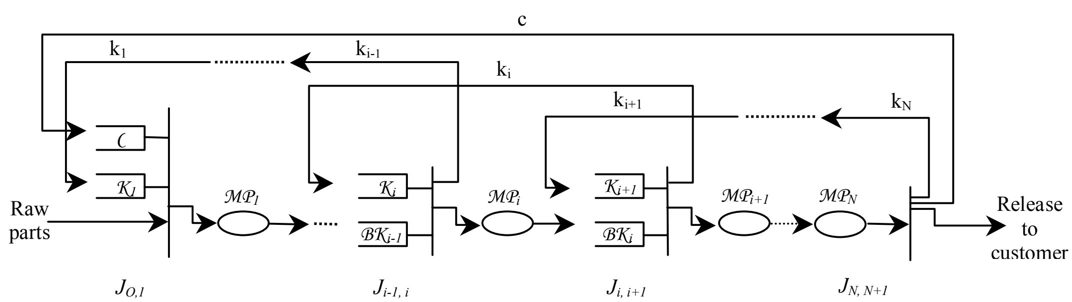


Figure 6-2: A saturated ECK control system having N stages in series.

Theorem 6.2 The production capacity of the ECK control system having $C \geq \sum_i k_i, i = 1,...,N$, depends only on $k_i, i = 1,...,N$, and is independent of $s_i, i = 1,...,N$.

Proof Since $C \geq \sum_i k_i, i = 1,...,N$, queue $\mathcal{C}$ plays no role in the synchronization station it belongs to, $J_{0,1}$; hence it can be eliminated. Also queue $\mathcal{K}_1$ will be the only queue feeding the first synchronization station; hence it can also be eliminated. Once queue $\mathcal{C}$ and $\mathcal{K}_1$ have been eliminated, the resulting queuing network is identical to the saturated queuing network of the simple kanban control system having N stages in series with k_i stage-i Kanbans.

This resulting queueing network is a basic FJQN/B containing N distinct elementary closed subnetworks, one in each stage, corresponding to the circulation loop of the kanban in that stage. Thus, the stage-i closed subnetwork has k_i customers. From Dallery *et al*. (1997), the throughput of a basic FJQN/B containing N distinct elementary closed subnetworks depend only on the fixed number of customers in each closed subnetwork – in this case k_i, $i = 1,\ldots,N$ – and not on the initial allocation of these customers among the different queues of this closed subnetwork. Therefore, the production capacity of the ECK control system having $C \geq \sum_i k_i, i = 1,...,N$ depends only on k_i, $i = 1,\ldots, N$ and is independent of s_i, $i = 1,\ldots, N$.

Property 6.3 The production capacity of the ECK system having $C \geq \sum_i k_i, i = 1,...,N$ is equal to the production capacity of the simple kanban control system with the same parameter k_i, $i = 1,\ldots, N$

Proof The saturated queuing network of the ECK system having $C \geq \sum_i k_i, i = 1,...,N$ and the saturated queuing network of SK control are similar. Even though the initial basestock level of them may be different (Dallery *et al.* 1997), the throughput depends only on the number of customers in each subnetwork, not its initial allocation which is equivalent to the basestock level.

Theorem 6.4 The production capacity of the ECK control system having $C \le k_i$, $i = 1,\ldots, N$ depends only on $\boldsymbol{C}$ and is independent of k_i, $i = 1,\ldots, N$ and s_i, $i = 1,\ldots, N$.

Proof From the saturated ECK control system shown in Figure 6-4, since $C \le k_i$, $i = 1,\ldots, N$, queue $\mathcal{K}_i$, $i = 1,\ldots, N$ play no role in the synchronization station they belong to; hence they can be eliminated. Once queues $\mathcal{K}_i$, $i = 1,\ldots, N$ have been eliminated, the resulting queuing network is similar to the saturated queuing network of the CONWIP control system having N stages in series with $\boldsymbol{C}$ CONWIP cards.

This resulting queuing network is a basic FJQN containing only one subnetwork with C customers. From Dallery *et al.* (1997), the throughput of a basic FJQN depends on the fixed number of customers in the system, which is C, and not on the initial allocation of these customers among different queues in this closed sub network. Therefore, the production capacity of the ECK system having $C \le k_i$, $i = 1,\ldots, N$ depends only on C and is independent of k_i, $i = 1,\ldots, N$ and s_i, $i = 1,\ldots, N$.

Theorem 6.5 The production capacity of the ECK system having $C \le k_i$, $i = 1,\ldots, N$ is equal to the production capacity of the CONWIP control system with the same amount of CONWIP, $\boldsymbol{C}$.

Proof same as Theorem 6.3

6.2.2 Production capacity of two-stage ECK control system having $\boldsymbol{k_1 = k_2 = k}$

In this section, we restrict our attention to a two-stage production case where the numbers of kanbans in each stage are identical, i.e. $\boldsymbol{K} = (k, k)$. Since we are interested in the production capacity, the basestock levels have no influence. So we have: $\boldsymbol{S} = (-, -)$. In this case, we have to resort to numerical experiments using equations found in Appendix B. We considered the cases where the numbers of kanbans and CONWIPs are 15, 17, 19, 30, 34, and 38. Note that the later three numbers are double of the first three numbers. This ensured that we covered the case where $k \leq C \leq 2{*}k$. Table 6.1 (A) gives the production capacity as function of $\boldsymbol{K}$ and $\boldsymbol{C}$. Table 6.1 (B) gives the simulation result to validate the analytical results. Each simulation ran for 10 replication having 110,000 time units with warm-up period of 10,000 time units in each replication. This parameter setting gives the 95% confidence interval of the average production rate to be less than 1% of its mean. Note that an upper bound of this production capacity is 1, since all station has same service rate of 1.

Table 6.1: Production capacity of two-stage ECK control system, $k_1 = k_2 = k$

(A) Analytical Results

k	C = 15	17	19	30	34	38	$(\mu_{max})_{Kanban}$
15	0.8333	0.8496	0.8623	0.8972	0.8972	0.8972	0.8972
17	0.8333	0.8500	0.8634	0.9018	0.9083	0.9083	0.9083
19	0.8333	0.8500	0.8636	0.9048	0.9120	0.9172	0.9194
30	0.8333	0.8500	0.8636	0.9091	0.9188	0.9262	0.9305
34	0.8333	0.8500	0.8636	0.9091	0.9189	0.9267	0.9416
38	0.8333	0.8500	0.8636	0.9091	0.9189	0.9268	0.9528
$(\mu_{max})_{CONWIP}$	0.8333	0.8500	0.8636	0.9091	0.9189	0.9268	

(B) Simulation Results

k	C = 15	17	19	30	34	38	$(\mu_{max})_{Kanban}$
15	0.8334	0.8489	0.8630	0.9026	0.9026	0.9026	0.9026
17	0.8334	0.8494	0.8642	0.9054	0.9129	0.9129	0.9129
19	0.8334	0.8494	0.8629	0.9076	0.9157	0.9219	0.9219
30	0.8334	0.8494	0.8629	0.9092	0.9189	0.9268	0.9490
34	0.8334	0.8494	0.8629	0.9092	0.9196	0.9270	0.9553
38	0.8334	0.8494	0.8629	0.9092	0.9196	0.9272	0.9591
$(\mu_{max})_{CONWIP}$	0.8334	0.8494	0.8629	0.9092	0.9196	0.9272	

In this case, the average error is 0.12% and maximum error is 0.6%. The production rates from analytical method are mostly lower than those obtained from simulation. The difference between the two tables increases as the system moves toward Kanban configuration, i.e. $C \rightarrow 2*k$. The last rows in (A) and (B) show the production capacity of a similar production system operating under CONWIP control policy having the same amount of CONWIP $\boldsymbol{C}$. The last column on the right of each table show the production capacity of a similar production system operating under Kanban control having the same amount of kanbans, $\boldsymbol{K}$.

The numbers between the two dashed lines represented the case where both CONWIP and kanban mechanisms are effective. The numbers on the right side of this region represents the case where only kanban mechanism is effective, i.e. $C \geq 2*k$, and the production rates of these cases are exactly the same as the production rate in the equivalent Kanban system shown in the last right column. Similarly, the numbers on the left side of this region represents the case where only CONWIP mechanism is effective, i.e. $C \leq k$, and the production rates of these cases are exactly the same as the production rate in the equivalent CONWIP system shown in the last row.

The numerical study shown in Table 6.1 indicates that CONWIP level have stronger effect on the production capacity than Kanbans. This result coincides with the state space analysis in Chapter 5. From state-space analysis, CONWIP level determines the total possible states which have more effect than the Kanbans which determine the maximum production rate of each stage.

In addition, we observed that CONWIP and Kanban have "limiting" effect on the other. That is, for a given value of $\boldsymbol{C}$ we can increase the production capacity by increasing the number of kanbans $\boldsymbol{K}$ until it reaches the upper bound the production capacity of a CONWIP system having $\boldsymbol{C}$ amount of CONWIP in the system. This is true also for given $\boldsymbol{K}$. Figure 6-3 shows the upper and lower bounds of this two-stage ECK control system.

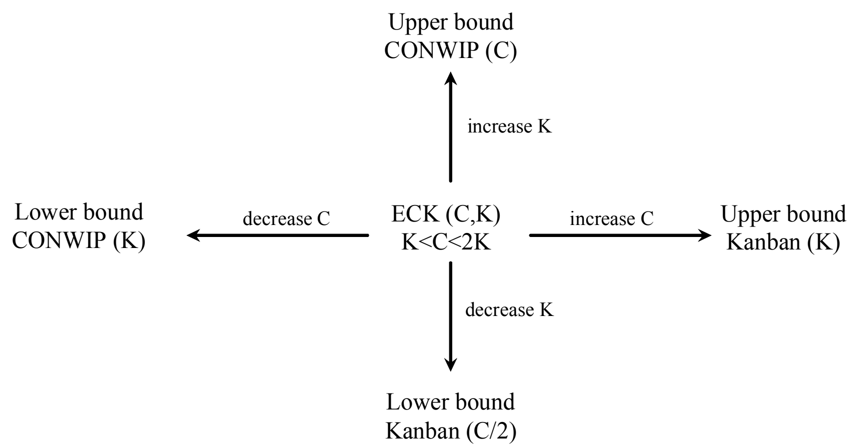


Figure 6-3: Bounds on the production capacity of ECK control system

The upper and lower bounds as a function of varying ***K*** is tighter than of varying ***C***. This gives rise that, it is better to use the bounds from varying ***K*** than varying ***C*** to predict the production capacity of a particular parameter set. For example, the production rate when ***C*** = 34 and ***K*** = 19 is 0.9120. The upper and lower bounds for the case of varying ***K***, i.e. vertical bounds, are 0.9189 and 0.9083 which is over five times tighter than that of varying ***C*** at 0.9172 and 0.8636.

The reasons why the case of varying ***K*** gives tighter bounds are as follow. Figure 6-3 shows two upper bounds due to CONWIP (***C***) and Kanban (***K***). Little's law implies that the system with higher amount of WIP will provide higher production rate. Under the saturated assumption, the amount of WIP in CONWIP (***C***) equals to ***C***, while the amount of WIP in Kanban (***K***) equals to $2*k$. When we consider $k \leq C \leq 2*k$, the production rate of saturated CONWIP (***C***) is always lower than or equal to Kanban (***K***). Hence, the upper bound of CONWIP (***C***) is always tighter. Similar reasoning can also be used for the lower bound case. The amount of WIP in saturated CONWIP (k) equals to k, while the amount of WIP in saturated Kanban (***C***/2) equals to 2*(***C***/2) = C. Since $\leq C \leq 2*k$, the production rate of saturated Kanban (***C***/2) is always higher than or equal to CONWIP (k); hence, the lower bound of Kanban (***C***/2) is always tighter.

Next, we study the effect when $k_1 \neq k_2$. From Table 6.1, in row k = 17, the range of C influencing production rate is between 17 and 34. Therefore the ranges of values we select are: $17 \leq k_i \leq 34$ and $17 \leq C \leq 34$. For reference purpose, the underlined values in Table 6.2 indicated those cases which have already been presented in Table 6.1

Table 6.2: Production capacity of two-stage ECK control system, $k_1 \neq k_2$

k_1	k_2	C							
		17	19	21	25	28	30	32	34
11	17	0.8467	0.8583	0.8674	0.8801	0.8866	0.8866	0.8866	0.8866
13	17	0.8489	0.8614	0.8711	0.8849	0.8921	0.8958	0.8958	0.8958
15	17	0.8498	0.8628	0.8731	0.8878	0.8954	0.8994	0.9028	0.9028
17	17	<u>0.8500</u>	<u>0.8634</u>	0.8741	0.8895	0.8975	<u>0.9018</u>	0.9053	<u>0.9083</u>
17	15	0.8498	0.8628	0.8731	0.8878	0.8954	0.8994	0.9028	0.9028
17	13	0.8489	0.8614	0.8711	0.8849	0.8921	0.8958	0.8958	0.8958
17	11	0.8467	0.8583	0.8674	0.8801	0.8866	0.8866	0.8866	0.8866

From Table 6.2, the values to the right side of the dashed line represent the cases where the number of CONWIPs plays no control role. The production capacity of the system is similar to those obtained from a similar system operating under Kanban policy with the same amount of kanbans. Therefore, we only focus on the values on the left side of the dashed line.

Another interesting observation in Table 6.2 is that the effect of k_1 and k_2 are symmetric, i.e. the production capacities decreases with k_1 and k_2 at the same rate. Our curiosity leads us to see what if we set the total amount of kanbans to be constant, i.e. $k_1 + k_2$ equals to 34. The results are shown in Table 6.3. The values to the left of dashed line in Table 6.3 represent the case where either the amount of stage-1 kanbans play no role (numbers in the lower left side) or the amount of stage-2 kanbans play no role (numbers in the upper left side). This indicate the capacity decreases with k_1 and k_2 from point (k_1, k_2) the slowest along the vector $k_1 = k_2$. As a result, the system with equals number of kanbans in both stages provide highest capacity.

Table 6.3: Production capacity of two-stage ECK control system, $k_1 \neq k_2$, $k_1 + k_2 = 34$

k_1	k_2	C							
		17	19	21	23	25	28	30	34
11	23	0.8467	0.8584	0.8678	0.8754	0.8815	0.8887	0.8925	0.8982
13	21	0.8489	0.8615	0.8716	0.8797	0.8863	0.8940	0.8980	0.9042
15	19	0.8498	0.8629	0.8735	0.8819	0.8887	0.8967	0.9009	0.9073
17	17	0.8500	0.8634	0.8741	0.8826	0.8895	0.8975	0.9018	0.9083
19	15	0.8498	0.8629	0.8735	0.8819	0.8887	0.8967	0.9009	0.9073
21	13	0.8489	0.8615	0.8716	0.8797	0.8863	0.8940	0.8980	0.9042
23	11	0.8467	0.8584	0.8678	0.8754	0.8815	0.8887	0.8925	0.8982

6.2.3 Effect of s_i on backorder and cycle time

The complexity now calls for simulation. Consider our two-stage production system with an external exponential demand process with rate 0.5. We first consider the effect of basestock on the backorders. We consider cases when $k_1 = k_2 = C = \{5, 10, 15\}$. This means that the kanban mechanism plays no significant role in this setting. The resulting ECK policy is therefore only the combination of CONWIP and Base Stock policy.

Intuitively, the basestock level of the last stage (s_2) is more important than the intermediate basestock level (s_1), so we set s_1 = {0, 5, 10, 15} and s_2 = {0, 5, 10, 15, 20}. The simulation results of percentage back-ordered demands as function of s_1 and s_2 are shown in Table 6.4. The number of replications is still 10 while each replication has the run length equals to 300,000 time units and warm up period equals to 30,000 time units. This longer simulation run was required in order to have the 95% confidence interval of the proportion of backorder demand to be within 10% of its associated mean value.

Table 6.4: Proportion of backorder demand of two-stage ECK control system, $k_1 = k_2 = C$

		s_1			
(C, k_1, k_2)	S_2	0	5	10	15
(5, 5, 5)					
	0	99.98%	99.98%	99.98%	99.98%
	5	44.67%	23.79%	11.90%	11.21%
	10	10.70%	5.07%	0.93%	0.66%
	15	2.52%	1.26%	0.09%	0.03%
	20	0.64%	0.38%	0.01%	0.00%
(10,10,10)					
	0	99.98%	99.98%	99.98%	99.98%
	5	36.86%	12.72%	11.72%	11.12%
	10	4.81%	0.87%	0.74%	0.62%
	15	0.45%	0.07%	0.05%	0.02%
	20	0.04%	0.00%	0.00%	0.00%
(15,15,15)					
	0	99.98%	99.98%	99.98%	99.98%
	5	36.51%	12.28%	11.43%	11.18%
	10	4.95%	0.84%	0.61%	0.63%
	15	0.40%	0.06%	0.03%	0.02%
	20	0.02%	0.00%	0.00%	0.00%

Table 6.4 suggests that both s_1 and s_2 have an influence on the proportion of back-ordered demands. However, the influence of s_2 is much stronger than s_1. In extreme case, when $s_1 = 0$ and $\boldsymbol{K}$ and $\boldsymbol{C}$ is such that the stability condition is verified (demand rate is less than production capacity), the proportion of backordered demands of this multi-stage system can be made as close as possible to 0 by increasing s_2. If s_2 is low; however, the proportion of backordered demands depend on the value of $\boldsymbol{K}$, $\boldsymbol{C}$ and s_1. This verifies the state-space analysis found in Chapter 5.

Table 6.5 shows total cycle time from the above simulations, i.e. production cycle time plus time waiting in finished goods inventory, of the same two-stage ECK system. The cases below dashed line indicate those that have backorder less than 5%. From these cases, the cycle time is more sensitive to s_2 when $s_1 < 10$ and it is more sensitive to s_1 when $s_1 \geq 10$. This implies that there exists an optimum s_1 such that increasing s_1 beyond this optimum level results only in increase WIP holding.

Table 6.5: Total cycle time of two-stage ECK control system, $k_1 = k_2 = C$

(C, k_1, k_2)	s_2	s_1 = 0	5	10	15
(5, 5, 5)					
	0	7.02	10.01	19.77	29.73
	5	10.01	15.17	25.98	35.98
	10	17.96	24.18	35.67	45.70
	15	27.52	33.98	45.66	55.69
	20	37.43	43.94	55.67	65.69
(10,10,10)					
	0	7.90	13.87	19.98	29.97
	5	11.34	19.98	26.19	36.24
	10	19.99	29.64	35.92	45.96
	15	29.84	39.61	45.89	55.94
	20	39.82	49.60	55.88	65.94
(15,15,15)					
	0	8.03	14.18	23.71	29.97
	5	11.47	20.31	29.96	36.23
	10	20.12	29.97	39.67	45.95
	15	29.97	39.94	49.64	55.92
	20	39.95	49.93	59.63	65.91

Finally, we studied when k_1 is not restricted to be equal to k_2. We consider the cases where the amount of CONWIPs equal to 10. We then set the values of k_1 and k_2 to be 5 or 10 and varied the value of s_1 and s_2 between 0 and 10 and between 10 and 15 respectively. Table 6.6 shows simulation results of the proportion of backordered demand.

Table 6.6: Proportion of backorder demand of two-stage ECK control system where k_1 and k_2 are independent

$C = 10$		$s_1 = 0$		$s_1 = 5$		$s_1 = 10$	
		$k_1 = 5$	$k_1 = 10$	$k_1 = 5$	$k_1 = 10$	$k_1 = 5$	$k_1 = 10$
$s_2 = 10$	$k_2 = 5$	4.956%	5.030%	0.982%	0.918%	0.765%	0.786%
	$k_2 = 10$	4.877%	4.811%	0.903%	0.867%	0.760%	0.744%
$s_2 = 15$	$k_2 = 5$	0.475%	0.468%	0.069%	0.058%	0.056%	0.076%
	$k_2 = 10$	0.478%	0.446%	0.070%	0.068%	0.051%	0.047%
$s_2 = 20$	$k_2 = 5$	0.050%	0.045%	0.003%	0.003%	0.007%	0.020%
	$k_2 = 10$	0.062%	0.038%	0.003%	0.004%	0.003%	0.002%

For reference purpose, the underlined values in Table 6.6 indicate the similar cases shown in Table 6.4. From Table 6.6, it is obvious that reducing either k_1 or k_2 causes increase in proportion of backorder demand. For the cases where s_2 is low, i.e. s_2 = 10, k_2 has stronger effect than k_1, while k_1 will have stronger effect when s_2 is large.

In conclusion, the production capacity seems to depend mainly of the number of CONWIP. Once the number of CONWIP is determined, varying the number of kanbans can be considered as a fine-tuning on the production capacity of the system. For a balance production line, it is desirable to assign similar number of kanbans to different stages if production capacity is the main concern. The effect of the basestock level of the last stage s_2 on proportion of back-ordered demand or the service level is stronger than that of intermediate basestock level s_1. However, when s_2 is not too low, reducing s_2 while increasing s_1 may yield more desirable result especially the case where the holding cost of finished goods inventory is higher than the holding cost of work-in-process. The effect of number of kanbans on service level is not significant.

6.3 Two-stage ECK control system separated by a bottleneck station

In general, every production line will have at least one bottleneck stage. Setting an inventory-cap makes sense if the selected section contains a bottleneck station. This is because releasing a part in an already congested section of the system with limited processing capacity will increase the inventory in that section with little or no decrease in the part's production time. It also makes sense to set some base stock in front of a bottleneck stage to ensure that this bottle neck stage will have very low starvation probability especially when the production lead time of the upstream stages is high or the demand process has high variability.

In this section, we would like to study the effects of $\boldsymbol{C}$, k_i, and s_i in a production line where there is one bottleneck station in it. Consider again our two-stage production system, but now with a service rate of station 3 equal to 0.8. The service rate of the other station remains 1. We chose this bottleneck station to be station 3 so that the effect of adding some inventory in front of the bottleneck station, which is our intermediate buffer, is magnified.

6.3.1 Production capacity of two-stage ECK control system with bottleneck station

We considered the same cases as in section 6.2.2 where the numbers of kanbans and CONWIPs are 15, 17, 19, 30, 34, and 38. Table 6.7 give the production capacity as function of $\boldsymbol{K}$ and $\boldsymbol{C}$. Note that an upper bound of this production capacity is obviously 0.8, since the bottleneck rate is equal to 0.8.

Table 6.7: Production capacity of two-stage ECK control system, $k_1 = k_2 = k$

k	C 15	17	19	30	34	38	$(\mu_{\max})_{\text{Kanban}}$
15	0.7574	0.7684	0.7761	0.7915	0.7915	0.7915	0.7915
17	0.7574	0.7686	0.7767	0.7935	0.7949	0.7949	0.7949
19	0.7574	0.7686	0.7769	0.7946	0.7962	0.7969	0.7969
30	0.7574	0.7686	0.7769	0.7959	0.7979	0.7989	0.7998
34	0.7574	0.7686	0.7769	0.7959	0.7979	0.7989	0.7999
38	0.7574	0.7686	0.7769	0.7959	0.7979	0.7990	0.8000
$(\mu_{\max})_{\text{CONWIP}}$	0.7574	0.7686	0.7769	0.7959	0.7979	0.7990	

From Table 6.7, the CONWIP level has stronger effect on the production capacity than kanbans, as in the case where there is no bottleneck station. Therefore, the upper and lower bounds as a function of varying ***K*** is tighter than that of varying ***C***.

Table 6.8 shows the production capacity of this two-stage ECK system with bottleneck station when we vary the amount of kanbans from 11 to 17 and the amount of CONWIP from 17 to 34. For reference purpose, the underlined values are the same as those presented in Table 6.7

Table 6.8: Production capacity of two-stage ECK control system with bottleneck station where $k_1 \neq k_2$

k_1	k_2	C 17	19	21	25	28	30	32	34
11	17	0.7677	0.7755	0.7811	0.7881	0.7910	0.7910	0.7910	0.7910
13	17	0.7683	0.7763	0.7819	0.7888	0.7917	0.7930	0.7930	0.7930
15	17	0.7686	0.7766	0.7823	0.7893	0.7921	0.7933	0.7941	0.7941
17	17	<u>0.7686</u>	<u>0.7767</u>	0.7825	0.7895	0.7923	<u>0.7935</u>	0.7943	<u>0.7949</u>
17	15	0.7684	0.7762	0.7816	0.7881	0.7906	0.7917	0.7924	0.7924
17	13	0.7675	0.7747	0.7797	0.7855	0.7877	0.7887	0.7887	0.7887
17	11	0.7651	0.7716	0.7759	0.7809	0.7828	0.7828	0.7828	0.7828

From Table 6.8, the effects of k_1 and k_2 are no longer symmetric. More precisely, k_2 now has more effect on the production capacity than k_1, i.e. the decrease in production rate by reducing the number of k_2 is greater than the decrease in production rate by reducing the same amount k_1, especially when $\boldsymbol{C}$ is high.

Table 6.9 shows the cases where we restrict the total amount of kanbans to be constant (equals to 34). Note that the values on the left of dashed line in Table 6.3 represent the case where either the amount of stage-1 kanbans play no role (numbers in the lower left side) or the amount of stage-2 kanbans play no role (numbers in the upper left side).

Table 6.9: Production capacity of two-stage ECK control system with bottleneck station where $k_1 \neq k_2$ and $k_1 + k_2$=34

k_1	k_2	C							
		17	19	21	23	25	28	30	34
11	23	0.7677	0.7756	0.7815	0.7859	0.7892	0.7927	0.7943	0.7963
13	21	0.7683	0.7764	0.7823	0.7867	0.7899	0.7931	0.7946	0.7964
15	19	0.7686	0.7767	0.7827	0.7869	0.7900	0.7930	0.7943	0.7959
17	17	0.7686	0.7767	0.7825	0.7866	0.7895	0.7923	0.7935	0.7949
19	15	0.7684	0.7762	0.7817	0.7855	0.7882	0.7907	0.7918	0.7930
21	13	0.7675	0.7748	0.7798	0.7833	0.7857	0.7879	0.7888	0.7899
23	11	0.7651	0.7716	0.7760	0.7790	0.7811	0.7830	0.7837	0.7846

From Table 6.9, by reducing k_1 while increasing the same amount of k_2, the production rate can be higher than the case where k_1 and k_2 are the same. It appears that the optimal numbers of k_1 and k_2 are now shifted from 17 and 17 as in the case where there is no bottleneck station to 13 and 21 respectively. Given the same number of

CONWIP, varying the number of kanbans from this optimal number always yields lower production capacity.

6.3.2 Effect of s_i in a multi-stage ECK control system with bottleneck station

In each case we assume that the numbers of kanbans and CONWIP in each stage are identical: five in the first case, ten in the second case, and fifteen in the third case. We varied the amount of s_1 from 0 to 15 while varied the amount of s_2 from 0 to 20. In table 6.10, we gave the proportion of back-ordered demands as function of s_1 and s_2.

Table 6.10: Proportion of backorder demand of two-stage ECK control system with bottleneck station where $k_1 = k_2 = C$

		s_1			
(C, k_1, k_2)	s_2	0	5	10	15
(5, 5, 5)					
	0	99.98%	99.98%	99.98%	99.98%
	5	57.39%	37.87%	20.71%	19.79%
	10	20.69%	11.87%	2.64%	2.32%
	15	6.75%	3.65%	0.30%	0.20%
	20	2.17%	1.03%	0.02%	0.02%
(10,10,10)					
	0	99.98%	99.98%	99.98%	99.98%
	5	45.03%	20.62%	19.94%	11.12%
	10	8.56%	2.57%	2.36%	0.62%
	15	1.22%	0.28%	0.25%	0.02%
	20	0.16%	0.04%	0.02%	0.00%
(15,15,15)					
	0	99.98%	99.98%	99.98%	99.98%
	5	36.51%	12.28%	11.43%	11.18%
	10	4.95%	0.84%	0.61%	0.63%
	15	0.40%	0.06%	0.03%	0.02%
	20	0.02%	0.00%	0.00%	0.00%

By comparing Table 6.9 with Table 6.4, it is obvious the amount of s_1 has more effect on the service level in the case where bottleneck station exists. For the same combination of number of CONWIP and kanban, the higher service level requirement, the higher s_1 is required in the system with bottleneck to achieve the same service level as in the case where there is no bottleneck.

Table 6.11: Proportion of backorder demand of two-stage ECK control system with bottleneck station where k_1 and k_2 are independent

$C = 10$		$s_1 = 0$		$s_1 = 5$		$s_1 = 10$	
		$k_1 = 5$	$k_1 = 10$	$k_1 = 5$	$k_1 = 10$	$k_1 = 5$	$k_1 = 10$
$s_2 = 10$	$k_2 = 5$	8.933%	9.040%	3.075%	2.912%	2.751%	2.562%
	$k_2 = 10$	8.693%	8.563%	2.594%	2.570%	2.359%	2.360%
$s_2 = 15$	$k_2 = 5$	1.297%	1.246%	0.437%	0.339%	0.349%	0.280%
	$k_2 = 10$	1.270%	1.217%	0.301%	0.284%	0.251%	0.246%
$s_2 = 20$	$k_2 = 5$	0.170%	0.116%	0.066%	0.039%	0.042%	0.030%
	$k_2 = 10$	0.194%	0.162%	0.041%	0.039%	0.033%	0.024%

From Table 6.11, k_2 has more effect on the service level at lower level of s_2, while k_1 has more effect on the service at higher level of s_2. In any cases, neither the numbers of k_1 or k_2 have much influence on the service level assuming that they are not too low.

In conclusion, the production capacity is mainly affected by the number of CONWIP. Once the number of CONWIP is determined, the number of kanbans at the bottleneck stage has more effect than that in the other stage. It appears that there exists an optimum value of kanban of the bottleneck stage such that the production capacity is maximize for any value of CONWIP.

The effect of the basestock level of the last stage s_2 on proportion of back-ordered demand or the service level is stronger than that of intermediate basestock level s_1. However, the intermediate basestock, s_1, has better impact to the service level than that in the balance production line case, i.e. it requires higher s_1 to achieve the same service level. The effect of number of kanbans on service level is not significant.

6.4 Design procedure for Extended CONWIP Kanban

Based on analysis in this chapter and in Chapter 5, the two most important parameters in the ECK system are the CONWIP level C and the finished goods basestock s_N. These two parameters have to be obtained first. The basestock levels at other stages, s_i, are the next important parameters since they directly affect WIP holding cost and also the service level. Finally, we search for the appropriate amount of kanbans at each stage, k_i, as a fine tuning on Inventory limiting at each stage.

In the first step, we treat the multi-stage ECK system as a single-stage ECK system and use algorithm in Section 6.1.3 to find the initial value of C and s_N, namely C^0 and s_N^0. Then, the maximum values for other parameters can be estimated based on these parameters. We set the maximum value of s_i equal to $h_i * s_N$ where hi equals to the ratio of finished goods holding cost to the WIP holding cost at stage i. For CONWIP value, we set the minimum value to be $C^0 / 2$ and the maximum value to be $2 * C^0$. These values can be set for a wider range.

The initial values of the k_i are set to be the same as C, so that the kanban mechanisms are not in effect at this step. We then enumerate all possible combinations of s_i and C to find the best combination which maximize the given performance evaluation.

Finally, we reduce the amount of kanbans from its initial value such that the kanban mechanism is now effective. We then enumerate all possible combinations of k_i to find the best final combination.

Step 1: Search for C^0 and s_N^0 using Algorithm 6.1

Step 2: Set $s_i^{max} = s_N^0$ for all i and $C^{min} = C^0 / 2$ and $C^{max} = 2 * C^0$

Step 3: Enumerate all possible combination of s_i and C given that $k_i = C$ for all i and find the best combination, s_i^*, C^*.

Step 4: Set $k^{max} = C^*$ and $k^{min} = 1$

Step 5: Enumerate all possible combination of k_i and find the best combination, k_i^*

Using the above algorithm is not guaranteed to obtain the optimum configuration; however, it is significantly easier to be implemented. In the first step, there are only two parameters to be specified regardless of the size of the production line. This type of initialization is not permitted in Kanban, Extended Kanban or Generalized Kanban. Once the first step is done, the performance of the system is guaranteed to be no worse than a CONWIP system.

The second step is to reduce the finished goods holding cost by reducing the finished goods basestock and increasing the intermediate basestock. The intermediate basestock is important when that buffer is followed by a bottleneck stage. Thus, system designer can reduce more computational time by considering only the basestock in front of the bottleneck stage and have all other basestock level to be zero.

Finally for the last step, one can study each stage separately and calculate the production capacity of that stage for different value of k_i. This value of k_i can be used as the minimum number of kanbans for that stage, k^{min}, which makes a tighter lower bound for the search algorithm.

CHAPTER 7

PARAMETER ADJUSTING MECHANISM FOR SINGLE-STAGE ECK POLICY

Pull systems are most successful in production environments with stable demand and lead times (Hall 1983). Unfortunately, systematic demand changes due to product cycle, natural and economic environment changes and inevitable. The pull system parameters derived from long-term averages are often misleading. Many companies periodically redesign their pull control mechanism to match current operating conditions. Other companies simply throw in additional cards or remove cards when they feel the need for more or less inventory. However, this is done in an *ad hoc* fashion, usually too late to be really beneficial, and often results in higher operating costs.

To achieve higher performance, system parameters have to be adjusted according to changes in demand. Most of the parameter adaptation mechanism focused on single parameter, single stage systems. We developed a procedure for adjusting a two-parameter according to changes in demand rate for a single-stage ECK control policy. The purpose of this chapter is not only to provide method for parameters adaptation but also to reveal the importance of adding a CONWIP control mechanism into Extended Kanban or Generalized Kanban policy.

7.1 Prior research

In this section, we summarize the adaptive control systems found in the literature. There are three types: Statistical Throughput Control, control chart-based reactive kanban control, and inventory-based adaptive kanban control. All of these are designed for single-stage kanban control system, which is equivalent to a CONWIP system, and have only one parameter to be controlled, the number of kanbans (or CONWIP).

7.1.1 Statistical throughput control

Hopp and Roof (1998) proposed a statistical throughput control that adjusts number of kanban cards by monitoring mean and standard deviation of inter-output times. Whenever average inter-output time is below (or above) the inverse of target throughput by more than three standard deviations, the system is considered out of control, or incapable to attain the throughput rate, and the number of kanbans is increased (or decreased) by one. Each time a card is added or subtracted, statistics are cleared and a warm-up period is observed before further decisions are made regarding number of kanbans.

7.1.2 Control chart-based reactive kanban control

Takahashi and Nakamura (1999) proposed a method that monitors time series of exponential smoothed inter-arrival times of demand and uses a control chart to alter the number of kanbans. The new number of kanbans in the system is determined using a control chart which is constructed *a priori* from simulation results.

7.1.3 Inventory-based adaptive kanban control

Tardif and Maaseidvaag (2001) proposed a method that monitors the inventory level and used to adjust the number of kanbans. It uses three supplemental inventory related parameters to achieve the control. First is the amount of extra cards available in the system. These extra cards can be released if the inventory level is too low. This implies that the maximum amount of parts in the system is equal to the amount of regular kanbans plus the amount of extra cards. Second parameter is the release threshold. When the inventory level is below the threshold an extra card, if any, is released into the system. The last parameter is the capture threshold. When the inventory level reaches this capture threshold a kanban card is captured and transferred to the extra card queue without triggering a release of new raw material into the system. When the demand arrival increases, which causes decreases in inventory level and increases in backorders, extra kanbans will be released into the system to accommodate this change. This higher amount of kanbans will increase throughput of the production system and the inventory level, hence expected backorders will decreased. When the surge in demand arrival has passed, extra kanbans will be captured from the system to prevent overstock of inventory.

7.1.4 Summary and design criteria of our control mechanism

In the Statistical Throughput Control, a target throughput has to be specified and the actual throughput of the system has to be monitored. This target throughput setting is not quite suitable for a pull production system; rather it is better coincided with the push philosophy. Also, numerous literatures indicate that controlling the throughput or cycle time is more difficult than controlling the amount of work-in-process or the finished goods inventory (Spearman *et al*. 1990, Spearmand and Zazanis 1992, Hopp and Spearman 1996). Therefore, one of our design criteria is to have a control mechanism that control the inventory level rather than throughput or cycle time.

In the control-chart based approach, the optimal number of kanbans for every possible demand rate has to be done off-line using simulation before the control decision can be made which may require enormous simulation runs. The simulation time requirement may cause this approach impractical. Hence, another design criteria for our control mechanism is that the control mechanism should not be based on any experimental simulation results; rather it should use only the information from the production process to adjust the control parameters.

In the inventory-based adaptive kanban control, the control system reacts to the changes in inventory level directly. This may result in over reacting to natural variation of the demand process instead of reacting only to the shifts in demand arrival rate. Also, there are three additional parameters to be specified which increases the complexity of the control method. Therefore, a demand detecting mechanism is needed to determine whether a real change in demand rate occurs.

Based on the summary of prior research, we came up the following design criteria.

1. Controls WIP or inventory level rather than throughput or cycle time
2. Does not based on experimental simulation results
3. Has a demand detecting mechanism to react only to real change in demand arrival rate

In addition, our single-stage ECK control system has two control parameters, namely number of CONWIP and the target basestock level. According to our best knowledge, there is no existing literature that purposes adaptive control on this two-parameter system.

7.2 Detecting changes in demand rate

We monitor the time series data of exponentially smoothed inter-arrival times of demand

$$H_i = \alpha \cdot x_i + (1-\alpha) \cdot H_{i-1}, \tag{7.1}$$

where H_i is the exponentially smoothed i^{th} inter-arrival time of demand, x_i is the original i^{th} inter-arrival time of demand, and α is the smoothing constant. Takahashi and Nakamura (1999) showed that α = 0.1 can detect the small unstable change with the least number of demands, and α = 0.3 can detect the large unstable change with the least number of demands. H_i will be compared to the upper and lower control limits. We consider system undergoes systematic change when the changes in demand inter-arrival rate are out of limits. The upper and lower control limits to detect unstable changes can be formulated as follows.

$$UCL = \mu_x + \delta \left(\frac{\alpha}{2-\alpha} \right)^{1/2} \sigma_x, \tag{7.2}$$

$$LCL = \mu_x - \delta \left(\frac{\alpha}{2-\alpha} \right)^{1/2} \sigma_x, \tag{7.3}$$

where δ is a multiplier and usually set as 3. By setting δ = 3, the probability that either the limits is violated without unstable changes is about 0.003. The performance of these limits can be found in Takahashi and Nakamura (1999).

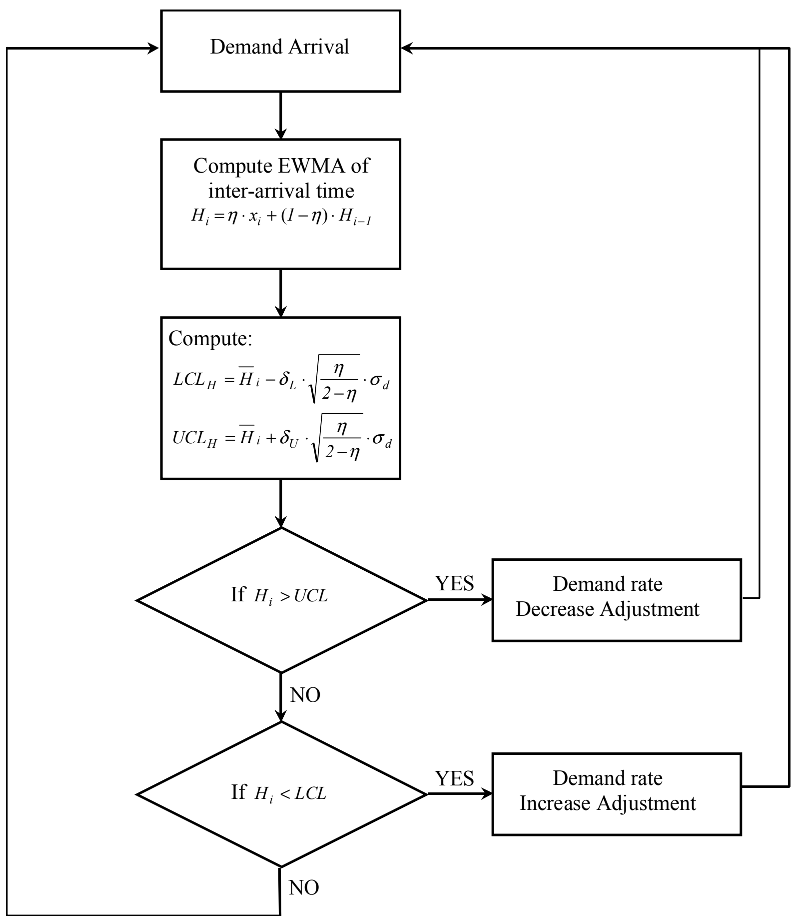


Figure 7-1: Demand detection method

7.3 Inventory level constraint

In this section we developed an approximation method in order to find the appropriate finished goods level once the demand rate has been changed. To do so, we first used normal approximation to estimate the basestock level in term of mean demand during production lead time and its variance. This gives the following result:

$$s = \upsilon + z \cdot \sigma_\upsilon \tag{7.4}$$

where s is the target basestock level

υ is the mean demand during lead time

σ_υ is the standard deviation of the demand during lead time

z is the standard normal coefficient

Using Little's law, the mean demand during lead time can be computed as follow:

$$\upsilon = \lambda_D \cdot E[LT] \tag{7.5}$$

where λ_D is the mean demand arrival rate

LT is production lead time

Let $E[BD(t)]$ be expected backordered demands at time t; $E[W(t)]$ be expected work-in-process at time t; $E[O(t)]$ be expected orders on-hold at time t; and $E[TH]$ be the expected throughput of the system. The expected production lead time can be expressed as follow:

$$E[LT] = \frac{E[W(t)] + E[O(t)] - E[BD(t)]}{E[TH]} \tag{7.6}$$

Assuming that the production line is always capable to serve the demand, i.e. the bottleneck rate is still higher than the demand rate, the expected throughput of the system is equal to the demand rate, i.e. $E[TH] = \lambda_D$. Equation (7.6) can now be expressed as follow:

$$\upsilon = E[W(t)] + E[O(t)] - E[BD(t)] \tag{7.7}$$

Using the property 3.5 from Chapter 3, the mean demand during lead time is now as follows:

$$\upsilon = s - E[I(t)] \tag{7.8}$$

By comparing Equation (7.4) and (7.8) , we have

$$E[I(t)] = z \cdot \sigma_\upsilon \tag{7.9}$$

Let the demand rate changed from λ_D to be $\alpha \cdot \lambda_D$, variance of demand during lead time will be

$$\begin{aligned} Var[\upsilon'] &= Var[\alpha \cdot \lambda_D \cdot E[LT']] \\ &= \alpha^2 \cdot Vare[\lambda_D \cdot E[LT']] \end{aligned} \tag{7.10}$$

We then approximated $E[LT']$ by $E[\alpha LT]$, which gives the following result:

$$\begin{aligned} Var[\upsilon'] &\approx \alpha^2 \cdot Var[\lambda_D \cdot E[k \cdot LT]] \\ &= \alpha^4 \cdot Var[\lambda_D \cdot E[LT]] \\ &= \alpha^4 \cdot Var[\upsilon] \end{aligned} \tag{7.11}$$

Thus,

$$\sigma_{\upsilon'} \approx \alpha^2 \cdot \sigma_\upsilon \tag{7.12}$$

Therefore, to accommodate the changes in demand arrival rate, the average finished goods level should be changed as follow:

$$E[I'(t)] = z \cdot \sigma_{v'} \approx \alpha^2 \cdot z \cdot \sigma_v = \alpha^2 \cdot E[I(t)] \tag{7.13}$$

Hence, in our feedback control, we need to monitor another time series: the level of finished goods:

$$I_i = \beta \cdot y_i + (1-\beta) \cdot I_{i-1}, \tag{7.14}$$

where I_i is the exponentially smoothed finished goods inventory level upon arrival of a demand, y_i is the original finished goods inventory level which is equal to actual amount of finished goods subtract amount of backorders, if any, and β is the smoothing constant.

7.4 Optimization formulation

The original optimization problem is

$$\text{Min } \text{Cost}(C,S) = h[E[W(t)] + E[FGI(t)]] - b \cdot E[BD(t)] \tag{7.15}$$

Where h is the holding cost per unit of work-in-process or finished goods

b is the backordering cost

The optimization problem in (7.15) can be rewritten as

$$\text{Min } \text{Cost}(C,S) = h[E[W(t)] + E[FGI(t)]] \tag{7.16}$$

$$\text{s.t. } P(I(t) > 0) \geq \frac{b}{b+h} \tag{7.17}$$

After the change in demand rate has been detected, we approximate the above optimization by

$$\text{Min } \text{Cost}(C,S) = h[E[W'(t)] + E[FGI'(t)]] \tag{7.18}$$

$$\text{s.t. } E[I'(t)] \geq k^2 \cdot E[I(t)] = \Omega \tag{7.19}$$

Since the cost function and $E[I'(t)]$ are increasing in C, the optimal value of C for a given s, C_s^*, is the smallest integer C that satisfies

$$E[I'_{C,s}(t)] \geq \alpha^2 \cdot E[I(t)] \qquad (7.17)$$

Which implies that

$$C_{s+1}^* \leq C_s^* \qquad \text{for } s \geq s_{\min} \qquad (7.18)$$

Where s_{min} is the optimal basestock level of the same system operating under pure Base Stock policy. The insight behind the above inequality is that C^* is non-increasing in C.

7.4.1 Demand rate changes

Here we develop a control method to adjust the finished goods inventory according to the changes in demand rate. Figure 7-2 shows the flow chart of our control method when demand rate decrease. Figure 7-3 shows the flow chart of our control method when demand rate increase.

For the case when demand rate decreases, we adjust the basestock level first while holding the same CONWIP level. Since the current CONWIP level can satisfy the demand rate before it changed, once the demand rate decrease this same CONWIP level can also satisfy the new demand rate. Once the basestock level has been adjusted, we reduce the CONWIP level by one unit such that the finished good inventory is just almost below the target value.

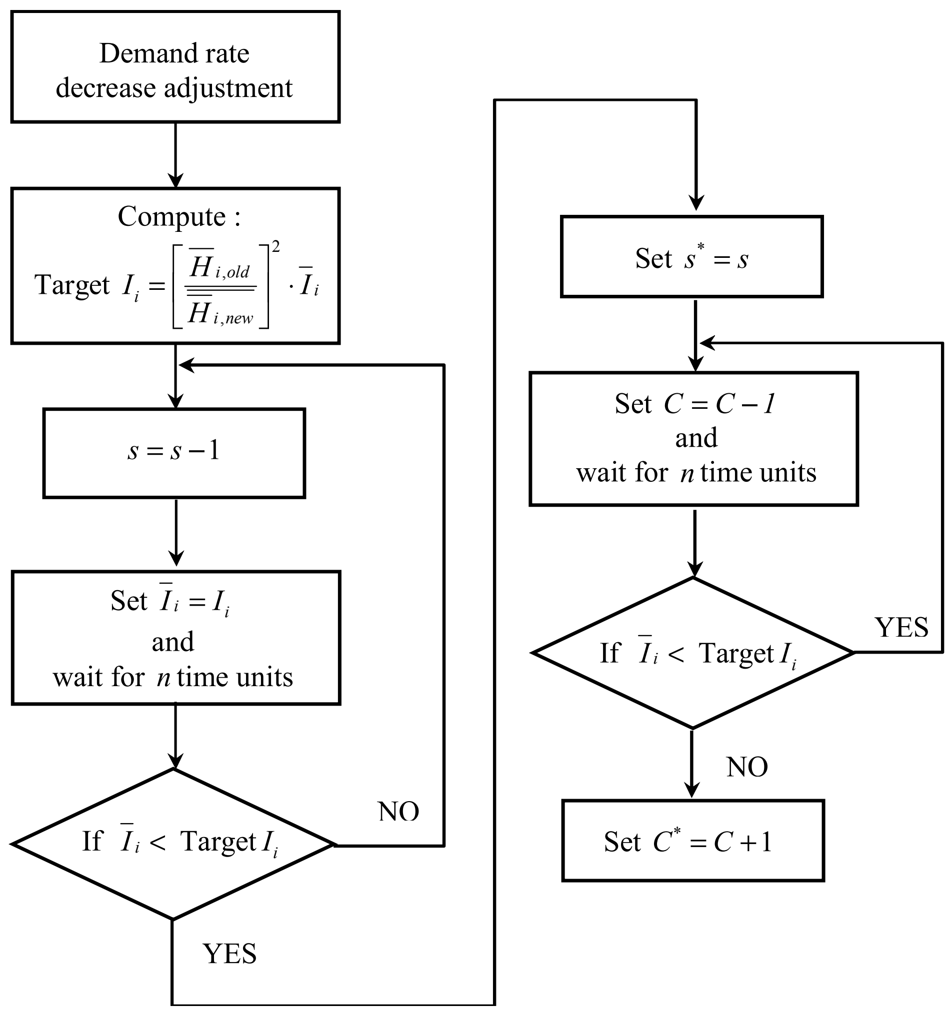


Figure 7-2: Flow chart of control method when demand rate decrease

For the case when demand rate increase, Figure 7-3, we increase both the basestock level and the CONWIP level until the finish goods target level is met. For fine tuning, we then reduce the CONWIP level by one unit such that the finished good inventory is just almost below the target value.

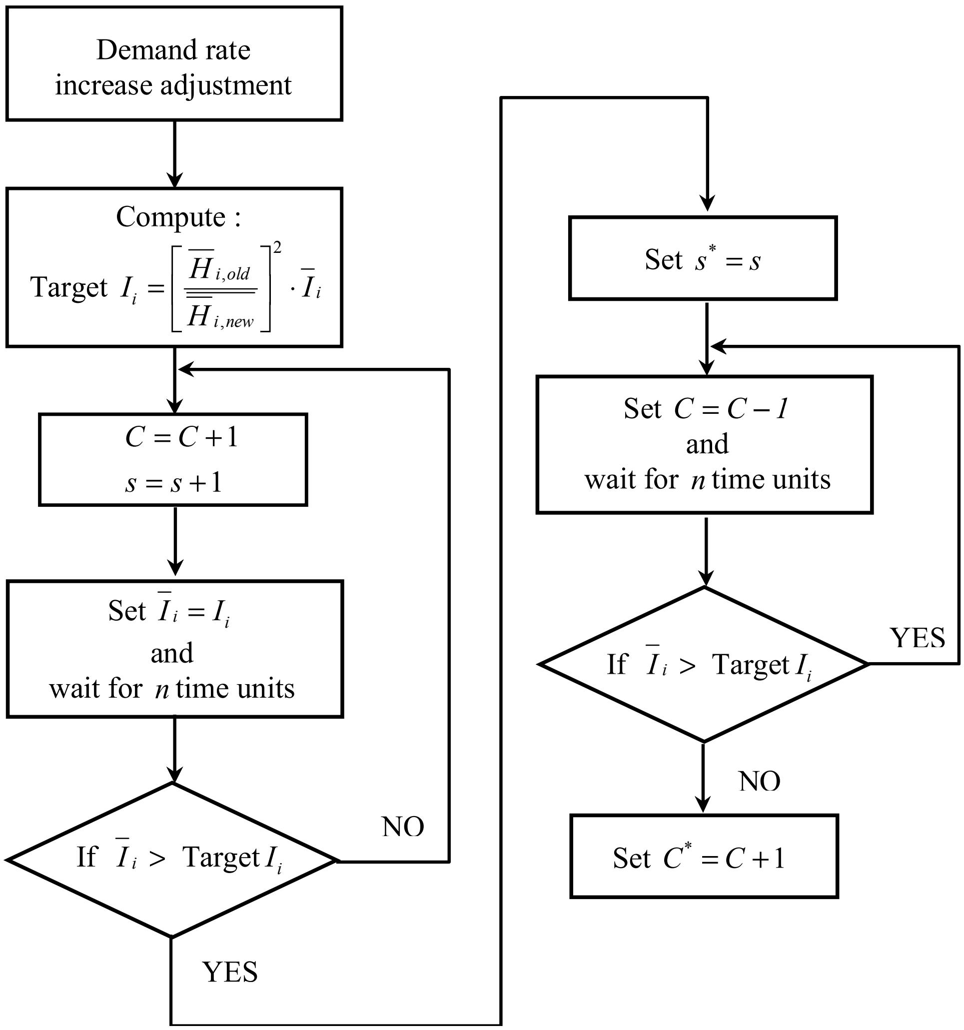


Figure 7-3: Flow chart of control method when demand rate increase

7.5 Performance of the purposed control mechanism

To illustrate the performance of our purposed control mechanism, we used simulated a single-stage ECK system containing four machine in tandem. We assumed that the WIP and finished goods holding cost is \$1 per time unit and the backordering cost is \$99 per time unit. The simulation run-length was 1,000,000 time units, excluding the warm-up run of 100,000 time units. The number of replications for each simulation run was ten replications. This yield 95% confidence intervals on the estimated values of $E[FGI]$, $E[BD]$, and $E[WIP]$ with half-width values of less than 0.5% of their respective estimated values. The distributions of inter-arrival time and processing time is shown in Table 7.1

Table 7.1: Configuration settings

	Processing Time		Inter-arrival time (before)		Inter-arrival time (after)	
	Distribution	Mean	Distribution	Mean	Distribution	Mean
Demand increase						
Case 1	GAMMA(2, 2.5)	5	Normal (7, 0.5)	7	Normal (6.5, 0.5)	6.5
Case 2	GAMMA(2, 2.5)	5	Expo (7)	7	Expo (6.5)	6.5
Case 3	Expo (5)	5	Normal (7, 0.5)	7	Normal (6.5, 0.5)	6.5
Demand decrease						
Case 1	GAMMA(2, 2.5)	5	Normal (7, 0.5)	7	Normal (7.5, 0.5)	7.5
Case 2	GAMMA(2, 2.5)	5	Expo (7)	7	Expo (7.5)	7.5
Case 3	Expo (5)	5	Normal (7, 0.5)	7	Normal (7.5, 0.5)	7.5

Figure 7-4-7-6 illustrates the exponentially smoothed finished goods inventory level as well as the corresponding CONWIP and basestock level for the cases when demand rate increases. Figure 7-7 – 7-9 illustrates the cases when demand rate decreases.

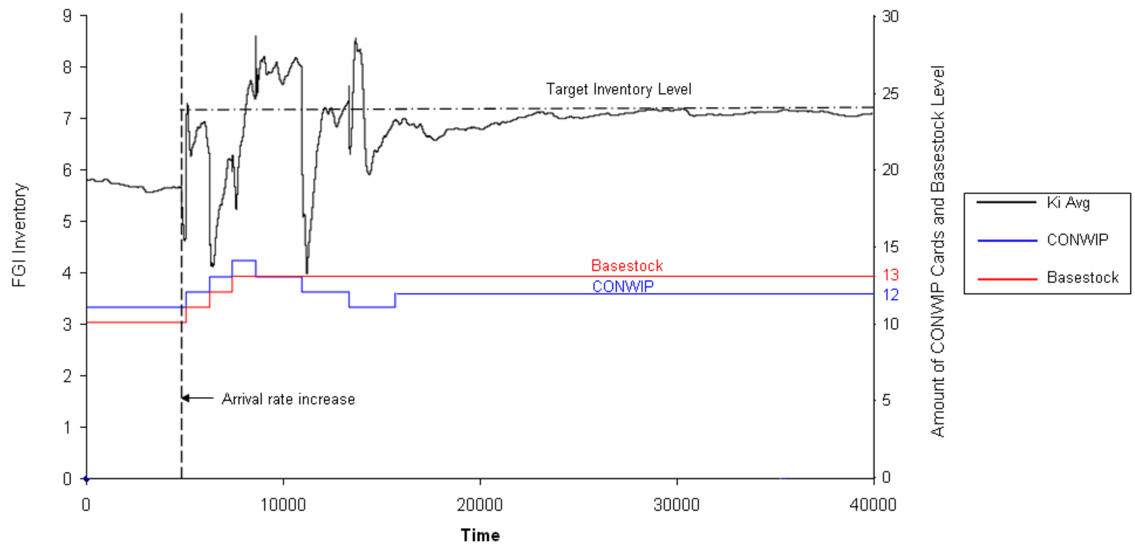


Figure 7-4: Demand rate increase, case 1

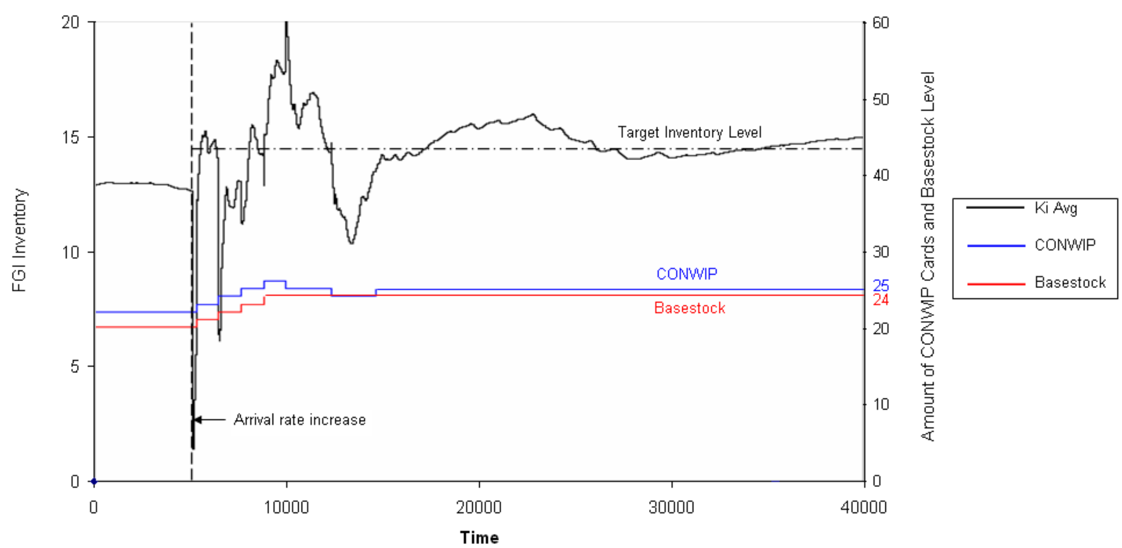


Figure 7-5: Demand rate increase, case 2

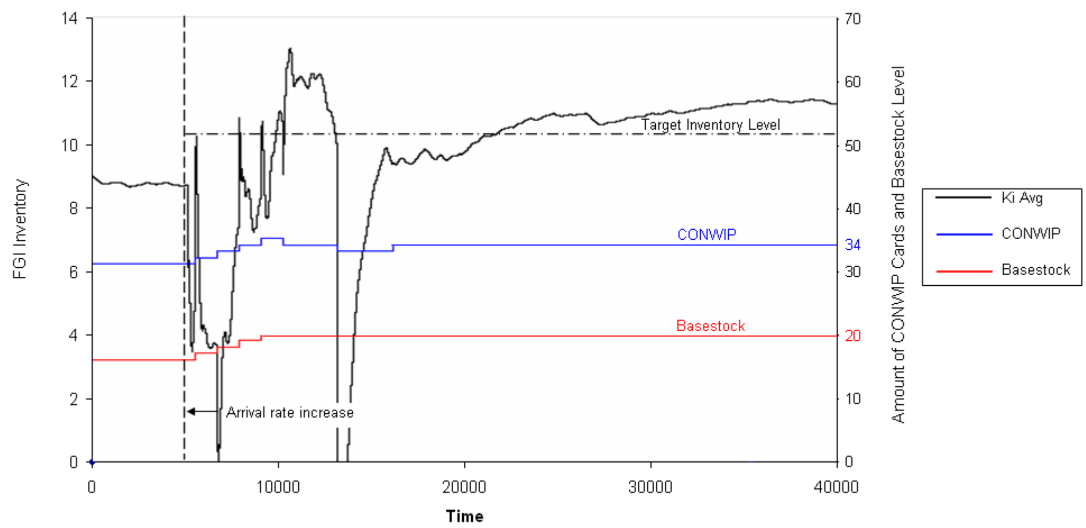


Figure 7-6: Demand rate increase, case 3

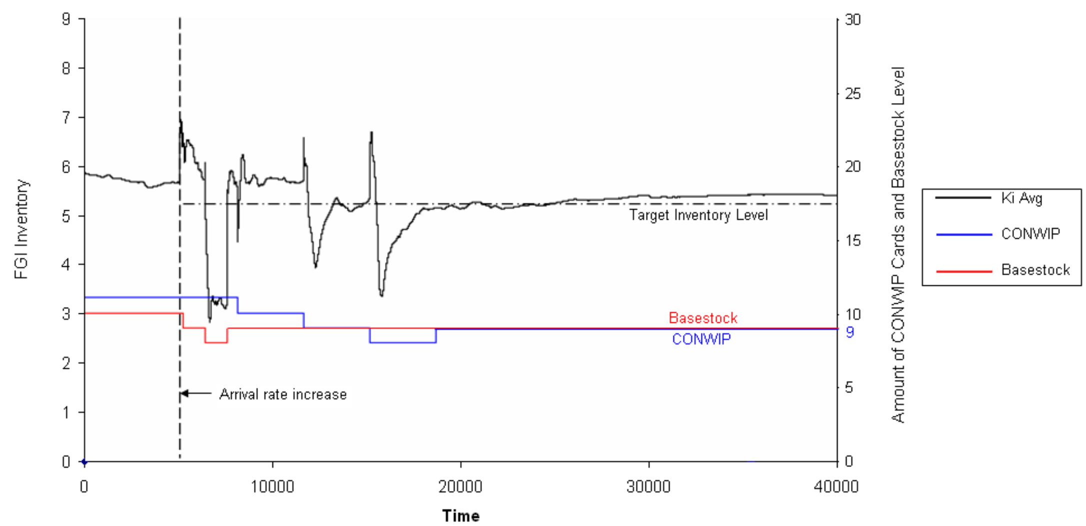


Figure 7-7: Demand rate decrease, case 1

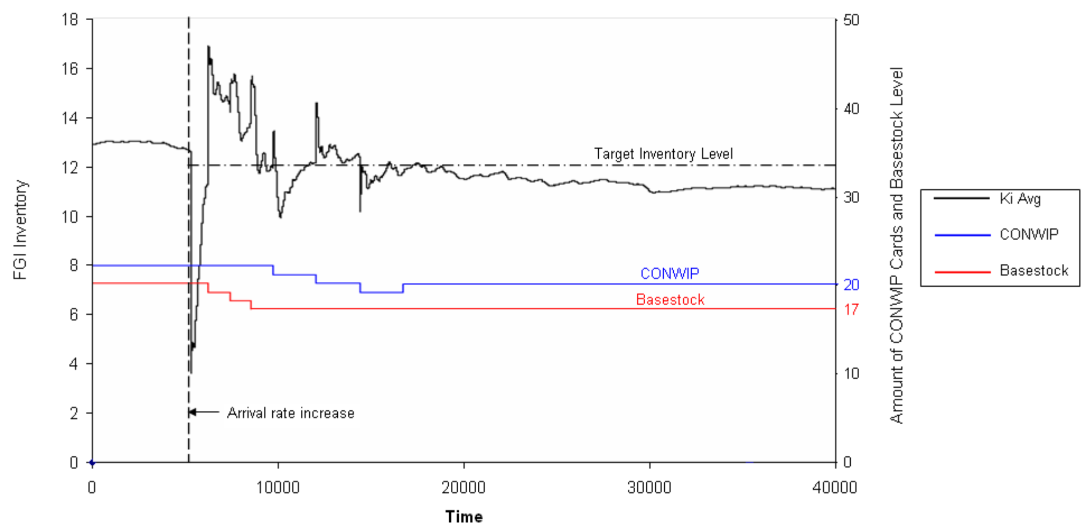


Figure 7-8: Demand rate decrease, case 2

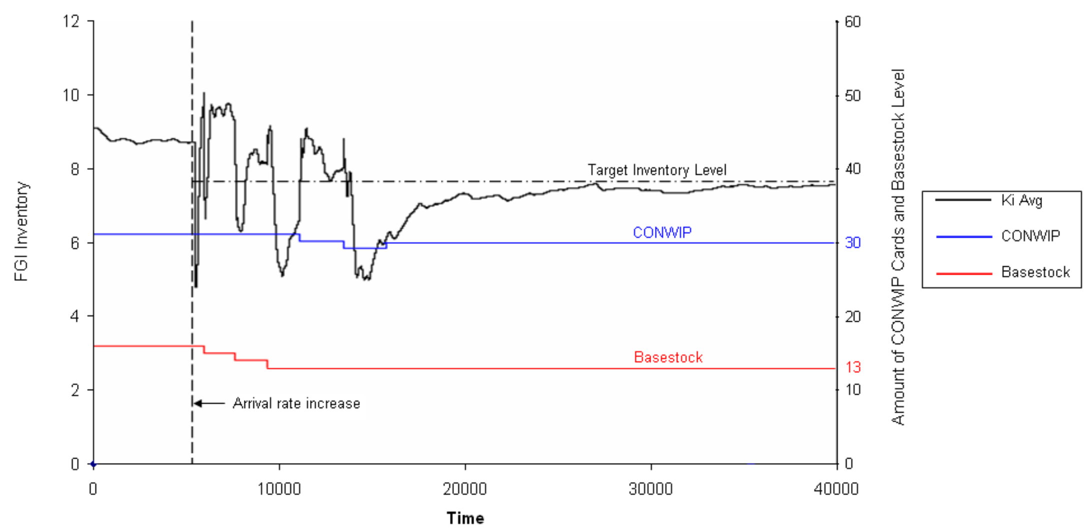


Figure 7-9: Demand rate decrease, case 3

Table 7.2 shows the results of our control mechanism compared with results from optimal configuration, which are obtained from brute-force simulation searches. We should point out that when we refer to simulation-based results, we use the word "optimal" for terminological simplicity but with caution, because we did not actually perform a significance test on the sign of the cost difference between two systems having different configurations. Using the word "near-optima" instead of "optimal" to describe the simulation-based results would be more accurate, but might be too wordy.

Table 7.2: Results of our control mechanism compared with optimal configuration

	Initial Setting		Ending Configuration			Optimal Configuration			% diff
	CW	BS	CW	BS	Cost	CW	BS	Cost	
Demand Increase									
Case 1	11	10	13	12	13.217	13	12	13.217	0.00%
Case 2	22	20	25	24	30.478	25	24	30.044	1.44%
Case 3	31	16	34	20	23.673	34	21	23.384	1.24%
Demand Decrease									
Case 1	11	10	9	9	9.179	11	8	8.625	6.43%
Case 2	22	20	20	17	19.748	18	17	19.730	0.09%
Case 3	31	16	30	13	14.584	30	13	14.584	0.00%

From Table 7.2, our control mechanism performance is within 1.5% percent in costs compared with optimum one, except one case where the difference is 6.43%.

7.6 Conclusion

We develop a simple adaptive production control method for setting CONWIP and basestock levels to meet changes in demand arrival rates in a pull production system operating under single-stage ECK policy. This method uses real-time data of demand inter-arrival time to detect if there are any changes in the mean arrival rate, then automatically adjust CONWIP and basestock level accordingly. Our control objective is to maintain the average amount of finished goods inventory at a pre-specified level obtained by the changes in mean inter-arrival time. Because this control objective is more practical than other control objectives, e.g. required service level or average amount of backlogs which can accurately be observed after a long period of time, it is well-suited to a real world situation especially when system is subjected to frequent demand change. Using simulation, we demonstrate the effectiveness of our control mechanism under a variety of conditions, including increase or decrease in demand rate and cyclical changes in demand rate.

CHAPTER 8

A CASE STUDY OF IRON AND STEEL PRODUCTION PLANT

This chapter presents a performance comparison of various control policies applied to an iron and steel production plant found in Hodgson and Wang (1991a, b). The control policies considered in this chapter included every control policy in Chapter 2 except Generalized Kanban control policy. A Markov Decision Process (MDP) is adapted and expanded to model each control policy using simulations. Detailed descriptions to the models of each control policy are presented in section 8.3. The experimental conditions and the validation of the simulation models against performance values in existing papers is presented in Section 8.4. The performances of each control policy are compared in Section 8.5 and 8.6. Then, they were compared with varying the mean demand rate in section 8.7. Finally, sections 8.8 and 8.9 provide a discussion and conclusions drawn from the results of the experimental process.

8.1 Description of the production system

Figure 8-1 shows the five stage parallel/serial line in Hodgson and Wang (1991a, b). It has two parallel stages following by three serial stages. One finished part from stage 1 and stage 2 are combined before entering stage 3. Demand in each period is either three or four units with fifty-fifty percent chance.

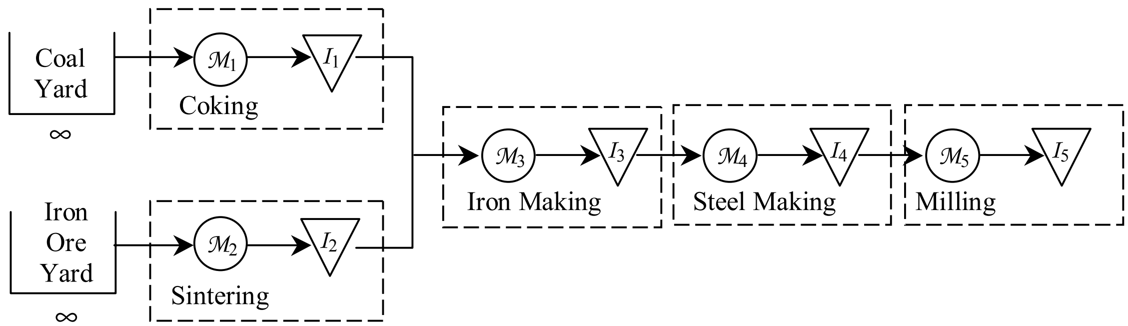


Figure 8-1: The Production Line of Iron and Steel Plant

Each machine has the same production capability and reliability with the production rate of either three, four or five units per period.

8.2 Modeling equations

That the structure of the system studied in this chapter is different from the simulation studies in chapter 4. Here, the production process is monitored in discrete time intervals, called production period, while the production process in chapter 4 is monitored in continuous time interval.

To determine the quantity to produce at a production stage in a given production period, we required solving two equations, which are production trigger and production objective. The production trigger is the amount of units the stage should produce in order to ensure that there will be enough units arrive to downstream stages to meet expected demands in subsequent periods. The production objective is the amount of units that the stage is capable of producing in the period given its production trigger, maximum production capacity, and the amount of units in its input and output buffers prior to the start of the production period. The production quantity is the amount of units the stage actually produces in the production period given its production objective and production reliability.

For the production reliability, the probability that stage j produces q units in period n given that the production objective is $PO_j(n)$, can be determined using the following equations:

For $q \in \{P_j^{\min},..., PO_j(n)-1\}$,

$$\Pr[q \mid PO_j(n)] = \Pr[P_j = q], \tag{8.1}$$

For $q \in \{PO_j(n),..., P_j^{\max}\}$,

$$\Pr[q \mid PO_j(n)] = \Pr[P_j = PO_j(n)] + \Pr[P_j = PO_j(n)+1] + ... + \Pr[P_j = P_j^{\max}] \tag{8.2}$$

The probability mass function for production reliability, $\Pr[P_j = q]$, is given in Table 8-1

Table 8.1: Probability mass function for production of each stage

q	3	4	5
$\Pr[P_j = q]$	0.2	0.6	0.2

In each production period, we assumed that the system calculated the production triggers and production objectives after the demands of that period had occurred and already consumed the finished goods inventory.

Let $A_j(n)$ = the production trigger for stage j in period n;

$\widetilde{D}(n)$ = the forecasted demand in period n;

$D(n)$ = the actual demand in period n;

$I_j(n)$ = the inventory in the output buffer of stage j in period n;

$I_j^{\max}$ = the maximum capacity of output buffer in stage j;

$P_j^{\max}$ = the maximum production rate of stage j;

$P_j^{\min}$ = the minimum production rate of stage j; and

ss = the desired safety stock level of the final product.

8.3 Descriptions of each control policy

Hodgson and Wang (1991a, b) observed that a strategy where the first two stages 'push' and all other stages 'pull' demonstrated the best result. Geraghty and Heavey (2004) compares the optimal hybrid push/pull control policy proposed by Hodgson and Wang (1991a, b) with hybrid CONWIP-Kanban control policy proposed by Bonvik *et al* (1997). The CONWIP-Kanban control policy is a control policy that incorporates both CONWIP and Kanban control mechanisms. The research showed that the optimal policy obtained in Hodgson and Wang (1991a, b) is equivalent to CONWIP-Kanban control policy proposed by Bonvik *et al* (1997).

8.3.1 Push control policy

For 'Push' control the production trigger for a stage, during period n, is given by:

$$A_5(n) = Max\{P_5^{\min}, ss + \widetilde{D}(n) - I_5(n-1)\},\tag{8.3}$$

$$A_4(n) = Max\{P_4^{\min}, ss + \widetilde{D}(n) - I_4(n-1) + \widetilde{D}(n) - I_4(n-1)\},\tag{8.4}$$

$$A_3(n) = Max\{P_3^{\min}, ss + \widetilde{D}(n) - I_5(n-1) + \widetilde{D}(n) - I_4(n-1) + \widetilde{D}(n) - I_3(n-1)\},\tag{8.5}$$

$$A_k(n) = Max\{P_k^{\min}, ss + 4 * \widetilde{D}(n) - I_5(n-1) - I_4(n-1) - I_3(n-1) - I_k(n-1)\}, k = 1,2.\tag{8.6}$$

The production objective of a stage, for period n, is given in Eqs. (8.7)-(8.10)

$$PO_5(n) = Min\{A_5(n), P_5^{\max}, I_4(n-1), I_5^{\max} - I_5(n-1) + \widetilde{D}(n)\},\tag{8.7}$$

$$PO_4(n) = Min\{A_4(n), P_4^{\max}, I_3(n-1), I_4^{\max} - I_4(n-1)\},\tag{8.8}$$

$$PO_3(n) = Min\{A_3(n), P_3^{\max}, I_1(n-1), I_2(n-1), I_3^{\max} - I_3(n-1)\},\tag{8.9}$$

$$PO_k(n) = Min\{A_k(n), P_k^{\max}, I_k^{\max} - I_k(n-1)\}, k = 1,2.\tag{8.10}$$

8.3.2 Kanban policy

In the Kanban control system, production at stage j is authorized when a Kanban is available and there is at least one part in the output buffer of the immediate upstream stage. A Kanban becomes available only when a finished part of the stage is transferred to the next or downstream stage.

The maximum capacity of the output buffer of stage j, I_j^{max} is equal to the amount of kanbans of that stage, i.e. $I_j^{max} = k_j$. Note that the Kanban control policy described here is equivalent to the pure pull system studied in Hodgson and Wang (1991 a, b). The production trigger for a stage, during period n, is given by:

$$A_j(n) = Max\{P_k^{\min}, k_j - I_j(n-1)\},\ j = 1,\ldots,J-1\tag{8.11}$$

$$A_J(n) = Max\{P_J^{\min}, ss - I_J(n-1) + \widetilde{D}(n)\}\tag{8.12}$$

The production objective of stage j, where $1 \leq j \leq J - 1$, is shown in Equation 8.13. The production objective for the final stage is obtained from Equation 8.14.

$$PO_j(n) = Min\{A_j(n), P_j^{\max}, \{I_{j-1}(n-1)\}, k_j - I_j(n-1)\}, \tag{8.13}$$

$$PO_J(n) = Min\{A_J(n), P_J^{\max}, \{I_{J-1}(n-1)\}, k_J - I_J(n-1) + \widetilde{D}(n)\}, \tag{8.14}$$

From the production triggers and production objectives defined above, the mechanism of the kanbans here is slightly different from the definition of kanban mechanism defined prior in that the number of kanbans available to the finial stage can be increased temporarily in response to a shortage. We used this definition of kanban mechanism so that we can validate our simulation models against results in existing paper.

8.3.3 CONWIP control policy

For CONWIP systems, production objective of every stage except the input stages (j = 1, 2) is to produce all of the available parts in its input buffer and is modeled by Equation (8.9). For the input stages, the production objective is also constrained by the amount of free CONWIP available and is shown in Equation (8.10).

$$PO_j(n) = Min\{P_j^{\max}, \{I_{j-1}(n-1)\}\}, \qquad j = 3, 4, 5 \tag{8.15}$$

$$PO_j(n) = Min\{P_j^{\max}, C - \sum_{i \in \{J\} - \{j'\}} I_i(n-1)\}, \qquad j = 1, 2 \tag{8.16}$$

8.3.4 CONWIP-Kanban control policy

Production triggers and production objectives of each stage in a CONWIP-Kanban system were determined by combining the equations used in the Kanban and CONWIP. The production trigger at each stage were modeled by Equation (8.17) and (8.18).

$$A_j(n) = Max\left\{P_j^{\min}, k_j - I_j(n-1)\right\}, \; k = 1,2,3,4, \tag{8.17}$$

$$A_j(n) = Max\left\{P_j^{\min}, k_j - I_j(n-1) + D(n)\right\}, \; k = 5 \tag{8.18}$$

For input stages of a CONWIP-Kanban system (j = 1, 2), the production trigger at each stage is modeled using Equation (8.20). This equation was developed by further constraining the production triggers in the Kanban system such that sufficient CONWIP must also be available at the stage to authorize the production. For other stages, j = 3, 4, 5, the production objectives are similar to those in the kanban system.

The production objective at each stage can be modeled as follow.

$$\begin{aligned} PO_5(n) &= Min\{A_5(n), P_5^{\max}, I_4(n-1), I_5^{\max} - I_5(n-1) + D(n)\} \\ &= Min\left\{P_5^{\max}, I_4(n-1), I_5^{\max} - I_5(n-1) + D(n)\right\} \end{aligned} \tag{8.19}$$

$$\begin{aligned} PO_j(n) &= Min\{A_j(n), P_j^{\max}, \{I_{j-1}(n-1)\}, k_j - I_j(n-1)\} \\ &= Min\left\{P_j^{\max}, \{I_{j-1}(n-1)\}, k_j - I_j(n-1)\right\}, \qquad j = 3, 4 \end{aligned} \tag{8.20}$$

$$\begin{aligned} PO_j(n) &= Min\{\, A_j(n),\, P_j^{\max}, k_j - I_j(n-1), C - \sum_{i \in \{J\}-\{j'\}} I_i(n-1) \,\} \\ &= Min\{\, P_j^{\max}, k_j - I_j(n-1), C - \sum_{i \in \{J\}-\{j'\}} I_i(n-1) \,\} \end{aligned} \tag{8.21}$$

8.3.5 Base Stock control policy

In the Base Stock controlled system, production at stage j in period n is authorized by the presence of demands transmitted to that stage. When a demand arrived, it is splitted and transmitted to each production stage to authorize the production at that stage.

When the stage begins production of a new part the demand at that stage is dropped. The number of demands at each production stage is dependent on the basestock and actual inventory of its downstream stages and can be computed as follow.

$$D_j(n) = \left[\sum_{i=j}^{J} \{ s_i - I_i(n-1) \} \right] + (J - j + 1) \cdot D(n), \qquad j = 3, 4, 5 \tag{8.22}$$

$$D_j(n) = \left[\sum_{i \in \{J\} - \{j\}} \{ s_i - I_i(n-1) \} \right] + 4 \cdot D(n), \qquad j = 1, 2 \tag{8.23}$$

The production trigger at each stage is modeled using Equation (8.24) and the production objective is modeled using Equation (8.25-8.26).

$$A_j(n) = Max\{P_j^{\min}, D_j(n)\} \tag{8.24}$$

$$PO_j(n) = Min\{A_j(n), P_j^{\max}, \{I_{j-1}(n-1)\}\}, \quad j = 3, 4, 5 \tag{8.25}$$

$$PO_j(n) = Min\{A_j(n), P_j^{\max}\},\ j = 1, 2 \tag{8.26}$$

8.3.6 Extended Kanban control policy

The production trigger for a stage is the same as those in Base Stock control policy, Eq. (8.22-8.24), while the production objective are the same as those in Kanban control policy. The production objective of a stage, for period *n*, is given in Eqs. (8.27) – (8.29).

$$PO_5(n) = Min\{A_5(n), P_5^{\max}, I_4(n-1), k_5 - I_5(n-1) + D(n)\} \tag{8.27}$$

$$PO_j(n) = Min\{A_j(n), P_j^{\max}, \{I_{j-1}(n-1)\}, k_j - I_j(n-1)\}, \quad j = 3, 4 \tag{8.28}$$

$$PO_j(n) = Min\{A_j(n), P_j^{\max}, k_j - I_j(n-1)\}, \quad j = 1, 2 \tag{8.29}$$

8.3.7 Extended CONWIP Kanban

The production trigger for a stage is the same as those in Base Stock control policy, Eqs. (8.22 - 8.24), while the production objective are the same as those in CONWIP-Kanban control policy. The production objective of a stage, for period *n*, is given in Eqs. (8.30) – (8.32).

$$PO_5(n) = Min\{A_5(n), P_5^{\max}, I_4(n-1), k_5 - I_5(n-1) + D(n)\} \tag{8.30}$$

$$PO_j(n) = Min\{A_j(n), P_j^{\max}, \{I_{j-1}(n-1)\}, k_j - I_j(n-1)\}, \quad j = 3, 4 \tag{8.31}$$

$$PO_j(n) = Min\{A_j(n), P_j^{\max}, k_j - I_j(n-1), C - \sum_{i \in \{J\}-\{j'\}} I_i(n-1)\}, \quad j = 1, 2 \tag{8.32}$$

8.4 Simulation model validation

A simulation model based on the MDP model proposed in Hodgson and Wang (1991a, b) was developed using simulation software package, Arena 5.0 from Rockwell Software Inc. The experiments in Hodgson and Wang (1991b) were replicated in order to validate our simulation models. In the validation experiments, the maximum inventory levels for each inventory point k_1, k_2, k_3, k_4 and k_5 were 9, 9, 9, 12 and 20, which are the same as those in the paper.

The forecasted demand, $\widetilde{D}(n)$, was assumed to be 4 units per production period and the minimum and maximum demand was 3 and 4 units, respectively with a probability of 0.5 each. Raw material was assumed to be available in sufficient quantities to meet the needs of input stages. Any unsatisfied demand at the end of a production period was backordered to the next production period and incurred a shortage cost of $30 per unit per time period. Inventory in the system, including raw material required for the period, was charged at $0.10 per unit. The gain, or cost, per period is the sum of inventory and shortage costs for the period.

There are eight cases to be verified, which are Push, Pull, and Hybrid Push-Pull systems. The Push system is modeled using the production triggers and production objectives defined in Equation (8.3) – (8.10). The Pull system here is equivalent to the Kanban system presented above and is modeled using the production triggers and production objectives defined in Equation (8.11) – (8.14). There are five cases of Hybrid Push-Pull policies which can be categorized by the merging point between Push and Pull mechanism. The production triggers and objectives of Hybrid Push-Pull are the combination of the ones from Push and Pull systems based on its location in the Hybrid system.

First, we attempt to validate our simulation model by comparing the simulation results from our model against what are reported in Hodgson and Wang (1991a, b) and Geraghty and Heavey (2004). These are listed in Table 8-2. Each simulation run composed of 50 replications having the run length of 10,000 time units with the warm up period of 1,000 time units.

Table 8.2: Comparison of results from model developed with others.

Policy	Control Policy	Simulation Result	Geraghty and Heavey		Hodgson and Wang		Average	Maximum
		Average Gain	Average Gain	% difference	Average Gain	% difference	% difference	% difference
1	Pull at all stages	3.90834	3.91012	0.05%	3.90588	0.06%	**0.03%**	**0.06%**
2	Push at all stages	3.13947	3.13964	0.01%	3.13101	0.27%	**0.14%**	**0.27%**
3	Push 1 & 2, Pull 3, 4 & 5	3.11186	3.11083	0.03%	3.10226	0.31%	**0.17%**	**0.31%**
4	Push 1, 2 &3 Pull 4 & 5	3.13389	3.13949	0.18%	3.13086	0.10%	**0.14%**	**0.18%**
5	Push 1, 2 & 4 Pull 3 & 5	3.11324	3.11134	0.06%	3.10277	0.34%	**0.20%**	**0.34%**
6	Push 3 & 4, Pull 1, 2 & 5	3.36406	3.36629	0.07%	3.36014	0.12%	**0.09%**	**0.12%**
7	Push 3, Pull 1, 2, 4 & 5	3.36165	3.36736	0.17%	3.35101	0.32%	**0.24%**	**0.32%**
8	Push 4, Pull 1, 2, 3 & 5	3.48922	3.48943	0.01%	3.48486	0.13%	**0.07%**	**0.13%**
Average % difference				**0.07%**		**0.20%**		
Maximum % difference				**0.18%**		**0.34%**		

The percent difference presented is computed using the following formula:

$$\%\, difference\ between\ A\ and\ B \quad = \quad \frac{\left[2\cdot\left|A-B\right|\right]}{\left[A+B\right]}\cdot 100\% \tag{8.33}$$

From Table 8-2, the averages of percent difference of the results from our simulation models compared with the results from Geraghty and Heavey (2004) and Hodgson and Wang (1991b) are 0.07% and 0.20% respectively, with the maximums of 0.18% and 0.34%. The averages of percent difference of each policy range from 0.03% to 0.24% with the maximums range from 0.06% to 0.34%. The majority of our results are closer to the results from Geraghty and Heavey (2004) than those from Hodgson and Wang (1991b). Since the percent difference in the results are very small, i.e. all are less than 0.34 percent, it is felt that the results from our simulation model are well replicating to those from Geraghty and Heavey (2004) and Hodgson and Wang (1991b).

8.5 Comparison results

In this section, we studied only card-based control policies studied throughout this thesis. We did not include Hybrid Push-Pull policy as part of this study. We also limited our study to the case where the WIP cap is static, i.e. WIP cap does not change dynamically, which is one of the case studied in Geraghty and Heavey (2004). The parameter settings for each control policy are presented in Table 8.3. The parameters used in Hybrid Push/Pull and CONWIP-Kanban policies are the same as those used in Hodgson and Wang (1991b) and Geraghty and Heavey (2004). For other policies, the parameters were set such that the performance of them were closest to the Kanban and CONWIP Kanban policies. This might not result in the best or optimal configuration for that policy

Table 8.3: Parameters setting

Control Policies	k_1	k_2	k_3	k_4	k_5	C	s_1	s_2	s_3	s_4	s_5
KB	9	9	9	12	8	N/A	N/A	N/A	N/A	N/A	N/A
CW	∞	∞	∞	∞	∞	24	N/A	N/A	N/A	N/A	N/A
CK	9	9	9	12	20	24	N/A	N/A	N/A	N/A	N/A
BS	∞	∞	∞	∞	∞	N/A	0	0	0	0	12
EK	9	9	9	12	20	N/A	0	0	0	0	11
ECK	9	9	9	12	20	24	0	0	0	0	12

The comparison results using the parameter configurations from Table 8.3 are shown in Table 8.4. It can be seen that CK and ECK policies provided minimal total gain with almost identical results, while Kanban policy was the worst performer, i.e. had maximal total gain. There was a large difference between Kanban policy and other control policies which was caused by a larger amount of WIP required. EK is the second worst performer in this case. CW and BS provided similar results and are better than EK but worse than CK and ECK. A small but significant difference in performance between

{CK, BS} and {CK, ECK} was noted; this was achieved by about 0.5% lower WIP required and about half of shortage incurred in {CK, ECK}.

Table 8.4: Comparison results

Control Policy	Average Number of				Cost			Service Level (%)	Cycle Time
	Raw Material	WIP	Total Inventory	Shortage	Inventory	Shortage	**Gain (Total)**		
KB	6.9992	31.6089	38.6081	0.00000	3.86081	0.00000	**3.86081**	100	7.5310
CW	6.9991	24.2110	31.2101	0.00008	3.12101	0.00240	**3.12341**	100	5.7910
CK	6.9986	24.0972	31.0958	0.00004	3.10958	0.00114	**3.11072**	100	5.7622
BS	6.9999	24.2557	31.2556	0.00018	3.12556	0.00534	**3.13090**	99.99	5.5335
EK	6.9999	26.3634	33.3633	0.00000	3.33633	0.00012	**3.33645**	100	6.2311
ECK	6.9985	24.0971	31.0956	0.00004	3.10956	0.00114	**3.11070**	100	5.7623

The last column in Table 8.4 compared the production cycle time of each policy. These production cycle times represented the times when a part has just been released from raw material buffers into the production line until it finished processing from the last stage and does not include the time it spends in the finished goods buffer. KB has longest production cycle time following by EK, CW, CK, ECK, and BS. The production cycle times of the policies that have CONWIP control mechanism, i.e. CW, CK and ECK, are very close to each other which is largely because of having the same amount of CONWIP cards. The production cycle time of BS is shortest which is not unexpected because once a part has been released in to the production line, there is no blocking mechanism to delay that part in between the line. In the EK system, the production cycle time was higher than the one in BS because of the kanban blocking mechanism in between the line. The comparison under optimal configuration is presented in the next section.

8.6 Comparison under optimal inventory and safety stock levels

The comparison we first find the optimal parameter setting of each control policy such that the inventory cost is minimized given that the service level is greater than 99.5%. In the literature, Hodgson and Wang (1991b) did not compare the different control policies under optimal inventory and safety stock levels. On the other hand, Geraghty and Heavey (2004) compare optimal Hybrid Push/Pull with CONWIP-Kanban under optimal parameter setting.

Geraghty and Heavey (2004) found that the range for the maximum inventory levels should be between 8 and 20 and the range for the safety stock level should be between 4 and 20. The values outside the above ranges either significantly degraded or did not improve the result for the gain. The results of their experiments showed that the optimal strategy for the Hybrid Push/Pull policy is the same optimal strategy found in Hodgson and Wang (1991b), i.e. stage 1 and 2 use push policy while the rest use pull policy, but with a lower gain. They also showed that the optimal CONWIP-Kanban policy found under optimal configuration provided the same result as optimal Hybrid Push/Pull policy.

The parameter settings for each control policy under optimal condition are presented in Table 8.5. The parameters used in Kanban and CONWIP-Kanban policies are the same as those used in Geraghty and Heavey (2004). For other control policies, we enumerated all possible combination using the bounds above and reported only the best configuration.

Table 8.5: Parameters setting under optimal condition

Control Policies	k_1	k_2	k_3	k_4	k_5	C	s_1	s_2	s_3	s_4	s_5
KB	8	8	9	9	8	N/A	N/A	N/A	N/A	N/A	N/A
CW	∞	∞	∞	∞	∞	20	N/A	N/A	N/A	N/A	N/A
CK	10	10	14	12	9	20	N/A	N/A	N/A	N/A	N/A
BS	∞	∞	∞	∞	∞	N/A	0	0	0	0	7
EK	10	10	14	12	9	N/A	0	0	0	0	7
ECK	10	10	13	11	9	20	0	0	0	0	8

The comparison results are shown in Table 8.6. Under optimal condition, ECK is still the best policy following by CK, CW, EK, BS and KB.

Table 8.6: Comparison results under optimal condition

Control Policy	Average Number of				Cost			Service Level (%)	Cycle Time
	Raw Material	WIP	Total Inventory	Shortage	Inventory	Shortage	**Gain (Total)**		
KB	7.0017	24.0608	31.0625	0.00613	3.10625	0.18396	**3.29021**	99.67	5.6673
CW	6.9998	20.2114	27.2112	0.00474	2.72112	0.14214	**2.86326**	99.67	4.7105
CK	7.0013	20.1822	27.1835	0.00478	2.71835	0.14328	**2.86163**	99.66	4.7046
BS	6.9999	22.2545	29.2544	0.00263	2.92544	0.07878	**3.00422**	99.83	5.0036
EK	7.0013	21.3543	28.3556	0.00317	2.83556	0.09516	**2.93072**	99.76	4.8583
ECK	6.9997	20.1824	27.1821	0.00458	2.71821	0.13746	**2.85567**	99.68	4.7059

Table 8.7: Percentage difference between result from ECK and others

Control Policies	Average Gain	Gain from ECK	% difference
KB	3.29021	2.85567	15.22%
CW	2.86326	2.85567	0.27%
CK	2.86163	2.85567	0.21%
BS	3.00422	2.85567	5.20%
EK	2.93072	2.85567	2.63%

From Table 8.7, results from CONWIP and CONWIP-Kanban are very close to the optimal result from Extended CONWIP Kanban. Gains from Kanban, Base Stock and Extended Kanban are significantly higher than that from the Extended CONWIP Kanban.

8.7 Sensitivity analysis

In this section we varied the mean demand rate from 3.5 ± 0.1 in 0.05 increments. The results presented in this section used the same configuration as those in optimal comparison in Section 8.7. The purpose of this study is to analyze the changes in the performances of each policy if the actual participating demand rate was not the same as the rate assumed when optimizing the parameter configuration. Table 8.8 and Figure 8-2 shows the changes in total cost when varying demand arrival rate. Note that in some graphs, extreme points in Kanban policy are off-the-chart.

Table 8.8: Numerical results of changes in total cost

Control Policies	Total cost when average demand arrival rate is				
	3.4	3.45	3.5	3.55	3.6
KB	3.21920	3.17674	3.28592	7.07273	147.31293
CW	2.71034	2.74510	2.86365	3.16695	3.84081
CK	2.69965	2.74743	2.86144	3.13151	4.02783
BS	2.90694	2.94065	3.01376	3.14143	3.68969
EK	2.83426	2.87484	2.94029	3.14470	3.72078
ECK	2.69863	2.74384	2.84853	3.12926	3.99405

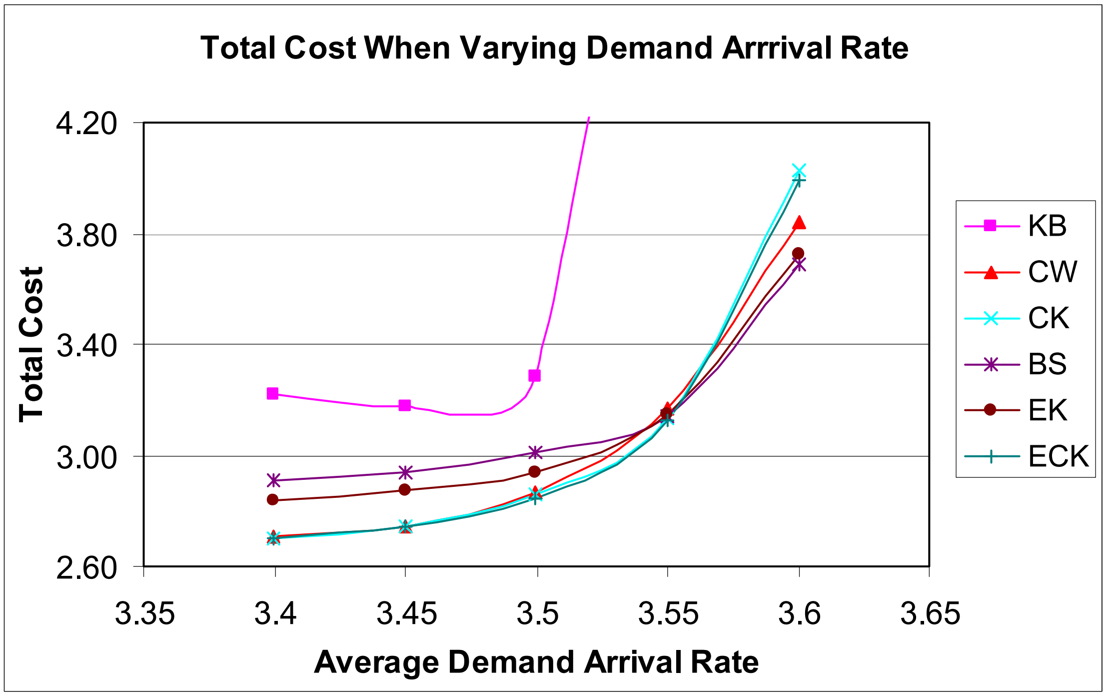


Figure 8-2: Changes in total cost when varying demand arrival rate

From Figure 8-2 and Table 8.8, it can be seen that the total cost of Kanban policy increased dramatically when the demand rate changed from its designed point. The total cost of all control policy, except Kanban policy, decreased when demand rate decreased and increased when demand rate increased. Extended CONWIP Kanban policy was the best performer here in every demand rate except when demand rate was at the highest value.

When demand rate was below the designed point, the performances of the policies with CONWIP mechanism were almost the same and were better then those without the CONWIP mechanism. The performance of EK was always better than the performance of BS and in between the BS and other policies with CONWIP mechanism. However, when the demand rate was above the designed point, the performances of the policies with

CONWIP mechanism deteriorated faster than those without one. At the highest demand rate considered in this study, the policies with CONWIP mechanism performed worse than those without one.

Therefore, for this production system, the Extended CONWIP Kanban is the most robust in terms of forecast errors. If the actual demand may be the same or lower than the estimated one, any of the policies that has CONWIP mechanism is appropriate. However, if the actual demand may be a lot higher than estimated, relaxing the CONWIP constraint and having either EK or BS control will perform better.

The total cost was calculated based on two components: the average WIP level and the amount of shortage. Table 8.9 and Figure 8-3 show the changes in average WIP level and Table 8.10 and Figure 8-4 shows the changes in average shortage.

Table 8.9: Numerical results of changes in average WIP level

Control Policies	Average WIP when demand arrival rate is				
	3.4	3.45	3.5	3.55	3.6
KB	25.3595	24.8220	24.0460	22.7076	19.2641
CW	20.1394	20.1743	20.2115	20.2623	20.3204
CK	20.1340	20.1621	20.1818	20.2039	20.2132
BS	22.1522	22.1983	22.2541	22.2834	22.3769
EK	21.4859	21.4002	21.3550	21.2910	21.2038
ECK	20.1337	20.1615	20.1828	20.2040	20.2135

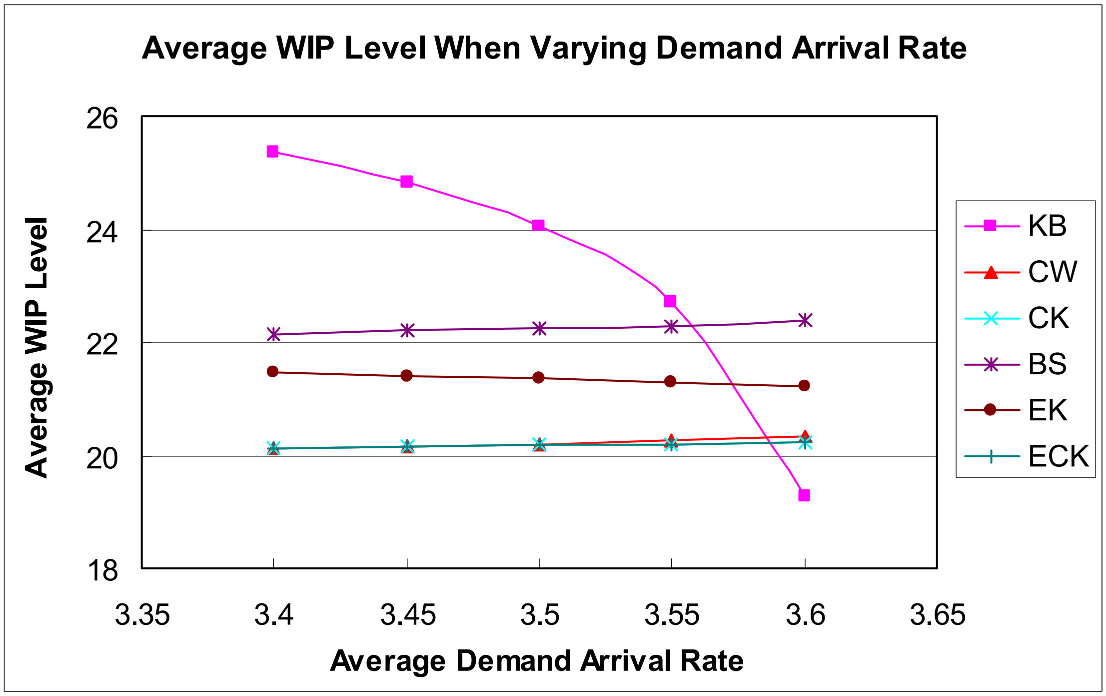


Figure 8-3: Changes in average WIP level when varying demand arrival rate

From Table 8.9 and Figure 8-3, every control policy, except KB, held fairly constant amount of WIP. When demand rate increased, only KB and EK policies had their average WIP level decreased. For KB, the WIP level in every stage decreased with a huge decrease in the last stage. This was primarily due to information flow delay. On the other hand, the EK and other policies had their WIP levels increased in every stage except the last one.

When demand rate decreased, only KB and EK policies had their average WIP level increased. For KB, the WIP level in every stage increased with a huge increase in stage 3. On the other hand, EK and other policies had their WIP levels decreased in every stage except the last one.

Therefore, the flow of information to the beginning of the production line, either through CONWIP mechanism or global demand mechanism, is essential to release sufficient new raw material into the production line.

Table 8.10: Comparison in average shortage

Control Policies	Average Shortage when average demand arrival rate is				
	3.4	3.45	3.5	3.55	3.6
KB	0.00010	0.00015	0.00603	0.13640	4.82220
CW	0.00055	0.00124	0.00475	0.01436	0.03629
CK	0.00021	0.00137	0.00477	0.01337	0.04287
BS	0.00040	0.00102	0.00294	0.00677	0.02439
EK	0.00019	0.00149	0.00349	0.01020	0.02935
ECK	0.00018	0.00125	0.00433	0.01330	0.04175

From Table 8.10 and Figure 8-4, for every policy the amount of shortage increased as demand rate increased and decreased as demand rate decreased. When demand rate decreased, the differences in amount of shortage in each policy are not significant. When the demand rate increase, since the amount of finished parts in KB policy decreased greatly, the shortage of KB increased dramatically and was not comparable to other policies. Among the other policies, BS policy which does not have any WIP limits performed best. EK policy, which is similar to BS policy but has local kanban limit, was the second best following by CW policy. CK and ECK policies, which have both CONWIP and kanban mechanism, performed worst excluding KB.

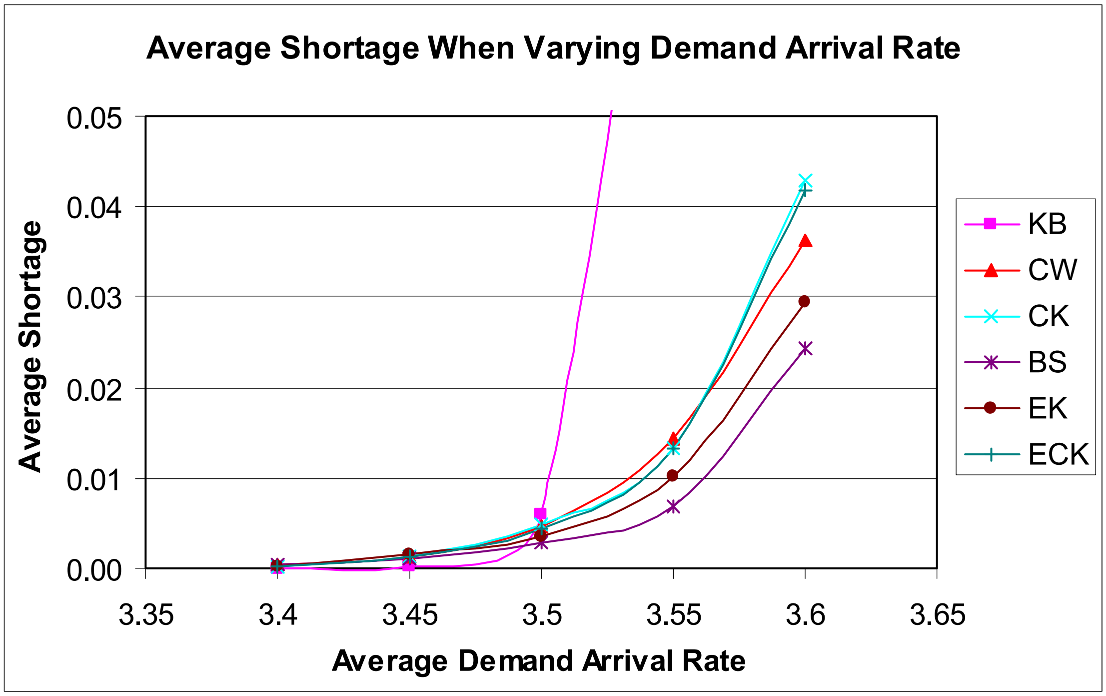


Figure 8-4: Changes in average shortage when varying demand arrival rate

Therefore, if the shortage cost is very high and the actual demand rate may be conservative, it is better to relax the CONWIP constraint, which is stricter than the kanban constraint, to allow more raw parts releasing into the system. The ECK policy can then be transformed to be the EK policy which can perform better if actual demand is higher than expected.

We then studied the effect of the changes in the variability of the performance measure. Table 8.11 and Figure 8-5 shows the changes in standard deviations of total cost when varying demand rate.

Table 8.11: Numerical results of changes in standard deviation of total cost

Control Policies	Standard Deviation of Total Cost when average demand arrival rate is				
	3.4	3.45	3.5	3.55	3.6
KB	0.0391699	0.0395643	0.0930335	4.1253758	57.8550441
CW	0.0373507	0.0437807	0.0997275	0.2659653	0.7411453
CK	0.0383020	0.0479347	0.1721075	0.3412010	1.2723498
BS	0.0462556	0.1205809	0.1587622	0.1964480	0.3914339
EK	0.0407659	0.0461708	0.1166725	0.1617132	0.5828739
ECK	0.0373990	0.0464497	0.1420212	0.2005785	0.7087420

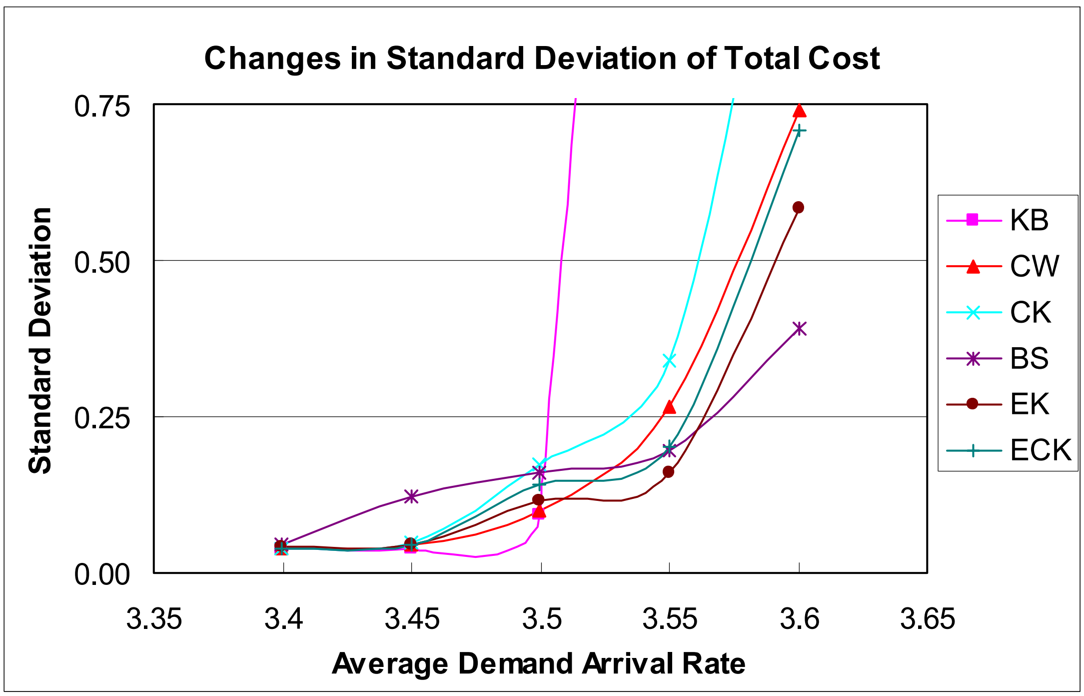


Figure 8-5: Changes in standard deviation of total cost

As in the case of the changes in the total cost, the changes in standard deviation of total cost in KB were a lot higher than other policies. The change in the standard deviation in BS was minimal and was due to its lower variation of shortage. One interesting result founded here is that the variation of the cost of ECK was lower than other policies with CONWIP mechanism. This is one of the advantages of introducing global demand flow mechanism in to the ECK policy over the other policies.

8.8 Discussion

Overall, Kanban policy was the least effective policy in addressing the WIP-service level trade-off. There was little difference between the other policies. The better performance of other policies over Kanban policy is due to the manner in which demand information is used by the policies. In a Kanban system demand information gradually propagates up-stream, one stage in each period, until it reaches the initial stages. The other policies instantly inform the initial stages of a demand event when it occurs. CONWIP and CONWIP-Kanban achieve this through the CONWIP mechanism while Base Stock, Extended Kanban and Extended CONWIP Kanban achieve it through the use of global demand information flow. Therefore, the poorer performance of Kanban policy relative to other policies is primarily due to information delay. This confirms the analysis and results in Chapter 4.

Another conclusion is that the policies that have CONWIP mechanism perform better than those without. Both Base Stock and Extended Kanban policies maintained more internal inventory and less external inventory than the other policies that have CONWIP mechanism. In the production system studied, the CONWIP mechanism plays an important role as an input controller that would not allow too much raw material releasing. The CONWIP mechanism utilizes the WIP information of the entire production line for this input control while the kanban mechanism in Extended Kanban policy uses only the local WIP information for input limiting.

While the improvement in performance over Kanban and Base Stock policies was dramatic, the improvement over CONWIP and CONWIP-Kanban policies was not distinguishable. One reason might be that the production system studied imposed minimum and maximum production quantity in each period which limited the range of the possible production quantity in each period to be narrow. Hence, the differences in the performance of different control policies were also narrow. Another reason was that the cost of holding unfinished parts inside the production line was the same as holding finished parts at the end of the line. This cost setting lessened the need for intermediate basestock inventories. Only input-output control mechanism for limiting the amount of WIP was important for this case. Therefore, the advantage of having basestock mechanism in ECK policy was accentuated.

8.9 Conclusion

In all cases, the ECK policy dominates all other policies. The reason is because it combines both the CONWIP mechanism and the global demand flow. Our sensitivity analysis also shows that ECK policy is more robust than other policy. Under inaccurate demand estimation case, even though the performance of ECK policy on the average cost is similar to those of the CONWIP and CONWIP-Kanban policies, its performance on the variation of cost outperforms that of other two policies. This confirms the most important advantage of ECK policy: it performs well under all circumstances.

CHAPTER 9

CONCLUSIONS AND FURTHER RESEARCH

9.1 Summary

In this thesis, we performed comprehensive study and analysis of existing pull control policies and have presented a new class of control policies for production system called Extended CONWIP Kanban policy (ECK). ECK is a hybrid of Base Stock, CONWIP and Kanban policies. It combines the advantages of the global information flow in Base Stock, the total WIP limiting in CONWIP, and the local WIP limiting in Kanban. Our analysis and simulation studies showed that the new hybrid policy is robust in varying situations and can outperform other policies including other existing hybrid policies with respect to attaining high service and throughput goals at low inventories.

Using a two-stage production system, we compared the characteristics of transition and transition rate boundaries of the optimum policy and various control policies in the state space. The analysis reveals that simple control policies such as CONWIP or Kanban policies is effective in some special production situations. However, these simple control policies do not have the flexibility to adjust their state spaces toward the optimum in general. This highlights the significant advantage of more complicated mechanisms such as the new hybrid policy. The ECK policy do not necessarily perform significantly better than the simpler ones for a given production situation but it is more robust under different circumstances, including changes in the system or demand variation and when the demand estimation is inaccurate.

Among the hybrid policies considered, ECK policy is the most flexible for adjusting the system characteristic. This conclusion from simulation study is confirmed by the results from state space comparison in Chapter 5. All of the transition boundaries, which are finished goods, WIP and buffer limits, can be independently adjusted and the transition rate boundaries can also be specified separately from the transition boundary. Finally, the maximum production capacity of each production stage can be limited individually.

On the design of system parameters, we identified the effect of each control parameters of at each stage in extreme situations and derived simple rules that can be used for designing multi-stage ECK system. We also purposed an adaptive parameter adjusting method for single-stage ECK system. This method adjusts both the finished goods basestock and the WIP limits. The results magnified the importance of being able to separately adjust the two parameters.

To verify that our result in 2-stage systems in more complex systems, we conducted a case study of iron and steel production plant. The performance of the ECK policy dominated all other policies regarding to cost minimization for a given service level. We also analyzed the robustness of each control policy by varying demand rates. The optimal parameter settings of each policy found under a given demand rate were tested under different demand rates. This reflects the case when demand estimation is inaccurate. The results show that Kanban policy was very sensitive to changes in demand rate and the performance of Kanban policy deteriorated dramatically. The changes in the total costs of ECK policy were very close to those of CONWIP and CONWIP-Kanban policies. However, by comparing the changes in standard deviation of total costs under different demand rates, the results shown that the variation in the total cost of the ECK policy was lower than those in the other two policies. This interesting result is very attractive in practice since the actual demand rate is always different from the estimated one.

9.2 Major contributions

In this dissertation,

- We provided a comprehensive anatomy of all pull type policies. We identified or documented the key invariants and bounds of these policies. The set provide the framework for the study.
- We developed a new control policy – ECK. We proved through simulation comparison and state space analysis that ECK policy perform better in any given situation and is more robust in varying situations.
- We developed a state space comparison methodology. Different pull policies (or any production control) operate similarly when there is no backlog, blockage or starvation. However, when they exist, the various policies perform differently. Our analysis can pin point these critical differences.
- In order to be practical and complete, we proposed parameter design and parameter adjustment for ECK policies. A case study is also included.

9.3 Further research

For performance comparison, a comparison of each control policy in a long production line could reveal some more interesting results. We believe that having the CONWIP mechanism in our ECK control would provide better performance due to the correlated WIP and could also help reduce the number of control points required in the system compared with other hybrid policies without the CONWIP mechanism. Additional research can then be in the area of finding the optimal number of control points. The performance comparison of a production line having correlated processing time would also be interesting.

For state-space analysis, it would be interesting to find the location of the optimal transition rate boundaries. Our analysis can explain only the flexibility of adjusting the transition rate boundaries. This can be achieved by optimizing the kanban parameters, since they define the maximum production capacity of each stage and are independent from other control parameters in the ECK system.

For the parameter design method, even though we had presented some simple rules for designing multi-stage ECK system, further works on the optimal design of ECK system are still needed. Specifically, the multi-stage case where the number of stages is more than two should be studied. The optimal design can be divided into two problems. The first one is the optimization of number of control points in the ECK system. In some circumstance, there might be no need to establish any control point in between the production line while it might be significantly better to have more than one intermediate control points. Once the number of control points in the system is determined, another important issue is the optimization of the parameters of the system. In addition, investigating on the issue of how to decompose the ECK system into smaller sub-systems in order to provide fast approximation on the performance of the system should be very useful for parameter design purpose. It would also be interesting to see if the conclusion drawn from our simple cases continue to apply for more complex production system such as those with assembly.

A fast approximation method to find the performance measure such as service level, average WIP and finished goods inventory for the ECK system would also be helpful in practice. Decomposition method used in Generalized Kanban is need to be explored.

The adaptive system concept proposed in this thesis is still in its inception. However, even with the relatively simple control rules, the system is able to perform well under changes in demand rate. More research is needed to further develop this concept. One task is to better assess the impact on multi-stage systems where there are more than one basestock level to be controlled. Another task is to incorporate the adjustment of kanbans in each stage; especially in the bottleneck stage. Also, as parameters in our system changed one by one, the amount changed of each parameter is a topic for further research. In addition, formal case studies of actual manufacturing system are needed to examine the results of implementing this adaptive parameter control in addition to implementing the ECK control policy.

For the iron and steel production case study, further research on the effect of the variation of demand distribution to the performance of each control policy could provided more evidence to support the robustness of ECK policy over other policy. Also, it might be interesting to see the comparison results when there is no minimum and maximum production quantity limits.

APPENDIX A

DERIVATION OF CONTROL FUNCTIONS

A 1 Control function of two-stage Base Stock policy

The control function of stage 1 is similar to the single-stage case except that the target basestock for stage 1 is equal to $s_1 + s_2$. Hence, the production rate out of stage 1 can be approximated as follow.

$$\mu_1(y_1) = \mu_1[s_1 + s_2 - (x_1 + x_2)] \tag{A.1}$$

For stage-2, the amount of parts being processed in stage 2, y_2, depends on the sum of x_1 and x_2 and can be characterized by the following relationship.

$$y_2 = \begin{cases} s_2 - x_2 & ; x_1 + x_2 \geq s_2 \\ x_1 & ; x_1 + x_2 < s_2 \end{cases} \tag{A.2}$$

Therefore, the production rate out of stage 2 can be approximated as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2(s_2 - x_2) & ; x_1 + x_2 \geq s_2 \\ \mu_2(x_1) & ; x_1 + x_2 < s_2 \end{cases} \tag{A.3}$$

A 2 Control function of two-stage Kanban policy

At stage 2, if the finished goods inventory position, x_2, is non-negative, the amount of parts in each in the intermediate synchronization station can be computed as follow.

$$NQ(\mathcal{B}) = x_1 - y_2 \tag{A.4}$$

$$NQ(\mathcal{K}_2) = k_2 - y_2 - x_2 \tag{A.5}$$

Since both of the queues are in the same synchronization station, the following equation must be true.

$$(x_1 - y_2)\cdot(k_2 - y_2 - x_2) = 0 \tag{A.6}$$

Using the fact that numbers of parts in each queue are non-negative, production rate out of stage 2 can be approximated as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2 - x_2) & ; x_1 + x_2 \geq k_2,\ x_2 \geq 0 \\ \mu_2(x_1) & ; x_1 + x_2 < k_2,\ x_2 \geq 0 \end{cases} \tag{A.7}$$

If the finished goods inventory is backordered, $x_2 < 0$, the sum of the free kanbans and the amount of parts in stage 2 is equal to the amount of stage-2 kanbans and the production rate out of stage 2 can be approximated as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ; x_1 \geq k_2,\ x_2 < 0 \\ \mu_2(x_1) & ; x_1 < k_2,\ x_2 < 0 \end{cases} \tag{A.8}$$

For the control function of stage 1, we again consider the relationships between queues in the intermediate synchronization station. However, the relationships have to be in term of y_1, which can be expressed as follow.

$$NQ(\mathcal{B}) = k_1 - y_1 \tag{A.9}$$

$$\begin{aligned} NQ(\mathcal{K}_2) &= k_2 - x_2 - y_2 \\ &= k_2 - x_2 - \left(x_1 - (k_1 - y_1)\right) \\ &= (k_1 + k_2) - (x_1 + x_2) - y_1 \end{aligned} \tag{A.10}$$

Using the property of the synchronization station, the following equation must be true.

$$(k_1 - y_1)\cdot(k_1 + k_2 - (x_1 + x_2) - y_1) = 0 \tag{A.11}$$

Using the fact that numbers of parts in each queue are non-negative, production rate out of stage 1 can be approximated as follow.

$$\mu_1(y_1) = \begin{cases} \mu_1[k_1 + k_2 - (x_1 + x_2)] & ; x_1 + x_2 \geq k_2, x_2 < 0 \\ \mu_1[k_1] & ; x_1 + x_2 < k_2, x_2 < 0 \end{cases} \tag{A.12}$$

When the finished goods inventory is backordered, the relationship regarding to the kanban control of stage 2 will be changed similar to previously mention argument and the production rate out of stage 1 can be approximated as follow.

$$\mu_1(y_1) = \begin{cases} \mu_1[k_1 + k_2 - x_1] & ; x_1 \geq k_2, x_2 < 0 \\ \mu_1[k_1] & ; x_1 < k_2, x_2 < 0 \end{cases} \tag{A.13}$$

A 3 Control function of two-stage CONWIP-Kanban control policy

To find the control function at stage 2, we analyzed the relationships between queues in the intermediate synchronization station, similar to the two-stage Kanban case, and obtained the approximation of the production rate out of stage 2 as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2 - x_2) & ; x_1 + x_2 \geq k_2, x_2 \geq 0 \\ \mu_2(x_1) & ; x_1 + x_2 < k_2, x_2 \geq 0 \end{cases} \tag{A.14}$$

And for the condition where the finished goods inventory is backordered, the production rate out of stage 2 is as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ; x_1 \geq k_2, x_2 < 0 \\ \mu_2(x_1) & ; x_1 < k_2, x_2 < 0 \end{cases} \tag{A.15}$$

To find the control function of stage 1, we analyzed the relationships between queues in the first synchronization station. First we consider the case where the finished goods inventory position is non-negative. The relationships between queues can be expressed as follow.

$$NQ(\mathcal{C}) = C - y_1 - (x_1 + x_2) \tag{A.16}$$

$$\begin{aligned} NQ(\mathcal{K}_1) &= k_1 - y_1 - NQ(\mathcal{B}) \\ &= k_1 - y_1 - (x_1 - y_2) \end{aligned} \tag{A.17}$$

$$(C - (x_1 + x_2) - y_1) \cdot (k_1 - x_1 + y_2 - y_1) = 0 \tag{A.18}$$

Using the fact that numbers of parts in each queue are non-negative, number of parts in production in stage 1 can be approximated as follow.

$$y_1 = \begin{cases} C - (x_1 + x_2) & ; x_2 - x_1 \geq C - k_1 - y_2,\ x_2 \geq 0 \\ k_1 - x_1 + y_2 & ; x_{21} - x_1 < C - k_1 - y_2,\ x_2 \geq 0 \end{cases} \tag{A.19}$$

Since there is the y_2 term in Equation (xxx), we substitute the value of y_2 from Equation (xxx) and obtained the following production rate out of stage 1.

$$\mu_1(y_1) = \begin{cases} \mu_1[C - (x_1 + x_2)] & ; x_1 + x_2 \geq C - k_1,\ x_2 \geq 0 \\ \mu_1(k_1) & ; x_1 + x_2 < C - k_1,\ x_2 \geq 0 \end{cases} \tag{A.20}$$

If the finished goods inventory is backordered, $x_2 < 0$, the production rate out of stage 1 can be obtained using similar method as in the case where $x_2 \geq 0$. The difference is that number of CONWIP is now equal to the sum of just y_1 and x_1.

$$\mu_1(y_1) = \begin{cases} \mu_1(C - x_1) & ; x_1 \geq C - k_1,\ x_2 \geq 0 \\ \mu_1(k_1) & ; x_1 < C - k_1,\ x_2 \geq 0 \end{cases} \tag{A.21}$$

A 4 Control function of two-stage Extended Kanban control policy

First, we analyzed the control function of stage 2. If the finished goods inventory position, x_2, does not fall k_2 units below the target basestock level, the kanban control of stage 2 plays no role, i.e. part authorization in the intermediate buffer will not be block because of no kanban. Therefore, we only consider queue $\mathcal{D}_2$ and queue $\mathcal{B}$ in the intermediate synchronization station. The relationships can be expressed as follow.

$$NQ(\mathcal{B}) = x_1 - y_2 \tag{A.22}$$

$$NQ(\mathcal{D}_2) = s_2 - x_2 - y_2 \tag{A.23}$$

$$(x_1 - y_2)\cdot(s_2 - x_2 - y_2) = 0 \tag{A.24}$$

Using the fact that numbers of parts in each queue are non-negative, the production rate out of stage 2, which is the same as production rate out of stage 2 in the two-stage Base Stock system, can be approximated as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2[s_2 - x_2] & ; x_1 + x_2 \geq s_2, x_2 \geq s_2 - k_2 \\ \mu_2[x_1] & ; x_1 + x_2 < s_2, x_2 \geq s_2 - k_2 \end{cases} \tag{A.25}$$

If the finished goods inventory position falls below $s_2 - k_2$, the amount of demands waiting in queue $\mathcal{D}_2$ is always positive; hence we would only consider queue $\mathcal{K}_2$ and queue $\mathcal{B}$ in the intermediate synchronization station and the relationships between queues can be expressed as follow.

$$NQ(\mathcal{K}_2) = k_2 - y_2 \tag{A.26}$$

$$NQ(\mathcal{B}) = x_1 - y_2 \tag{A.27}$$

Using the property of the synchronization station and the fact that both queues are non-negative, the production rate out of stage 2 can be approximated as follow:

$$\mu_2(y_2) \quad = \quad \begin{cases} \mu_2(k_2) & ; x_1 \ge k_2,\, x_2 < s_2 - k_2 \\ \mu_2(x_1) & ; x_1 < k_2,\, x_2 < s_2 - k_2 \end{cases} \qquad \text{(A.28)}$$

To analyze the control function of stage 1, the first condition is whether or not the number of free stage-1 kanbans is zero. The number of free stage-1 kanbans is zero when the surplus inventory from stage 1 is below the target surplus level by more than the sum of the total kanbans, i.e. $x_2 < (s_1 + s_2) - (k_1 + k_2)$. Using this fact, the relationships between queues in the intermediate synchronization station can be expressed as follow.

$$NQ(\mathcal{B}) \quad = \quad k_1 - y_1 \qquad \text{(A.29)}$$

$$\begin{aligned} NQ(\mathcal{K}_2) &= k_2 - y_2 \\ &= k_2 - (x_1 - NQ(\mathcal{B})) \\ &= (k_1 + k_2) - x_1 - y_1 \end{aligned} \qquad \text{(A.30)}$$

The condition that $x_2 \le (s_1 + s_2) - (k_1 + k_2)$ also satisfies $x_2 \le s_2 - k_2$ since the Extended Kanban policy requires $s_1 \le k_1$. Thus, when $x_2 \le (s_1 + s_2) - (k_1 + k_2)$ queue $\mathcal{D}_2$ is always positive, similar to the argument when we analyze the control function of stage 2, and the relationship in the intermediate synchronization station can be expressed as follow.

$$(k_1 - y_1)\cdot(k_1 + k_2 - x_1 - y_1) = 0 \qquad \text{(A.31)}$$

Using that fact that both queues are non-negative, the production rate out of stage 1 can be approximated as follow.

$$\mu_1(y_1) \;=\; \begin{cases} \mu_1[k_1 + k_2 - x_1] & ; x_1 \ge k_2,\, x_2 < (s_1 + s_2) - (k_1 + k_2) \\ \mu_1[k_1] & ; x_1 < k_2,\, x_2 < (s_1 + s_2) - (k_1 + k_2) \end{cases} \qquad \text{(A.32)}$$

For the control function of stage 1 when $x_1 \ge (s_1 + s_2) - (k_1 + k_2)$, we analyzed the number of parts in queues in the first synchronization station, which are as followed.

$$NQ(\mathcal{K}_1) \quad = \quad k_1 - y_1 - NQ(\mathcal{B})$$

$$= \quad k_1 - x_1 + y_2 - y_1 \tag{A.33}$$

$$NQ(\mathcal{D}_1) \quad = \quad (s_1 + s_2) - (y_1 + x_1 + x_2)$$

$$= \quad (s_1 + s_2) - (x_1 + x_2) - y_1) \tag{A.34}$$

Using the fact that numbers of parts in each queue are non-negative, number of parts in production in stage 1 can be expressed as follow.

For $x_2 \geq (s_1 + s_2) - (k_1 + k_2)$;

$$y_1 = \begin{cases} (s_1 + s_2) - (x_1 + x_2) & y_2 \geq (s_1 + s_2) - k_1 - x_2 \\ k_1 - x_1 + y_2 & y_2 < (s_1 + s_2) - k_1 - x_2 \end{cases} \tag{A.35}$$

Substituting the value of y_2 from Equation (xxx) – (xxx), the production rate out of stage 1 can be approximated as follow.

For $x_2 \geq (s_1 + s_2) - (k_1 + k_2)$;

$$\mu_1(y_1) = \begin{cases} \mu_1[(s_1 + s_2) - (x_1 + x_2)] & ;(x_1 + x_2) \geq s_1 + s_2 - k_1 \\ \mu_1[k_1] & ;(x_1 + x_2) < s_1 + s_2 - k_1 \end{cases} \tag{A.36}$$

A 5 Control function of two-stage Generalized Kanban control policy

We first analyzed the control function of stage when the finished goods inventory position, x_2, does not fall k_2 units below the target basestock level. In this case, the kanban control of stage 2 plays no role, i.e. queue $\mathcal{K}_2$ will always have parts in it and queue $\mathcal{D}_2$ is always empty. Therefore, we only have to analyze the relationship between the numbers of parts in queue $\mathcal{DK}_2$ and queue $\mathcal{B}$. When $x_2 < s_2 - k_2$, queue $\mathcal{D}_2$ will always positive and queue $\mathcal{K}_2$ is always empty; hence we again only have to analyze the relationship between queue $\mathcal{DK}_2$ and queue $\mathcal{B}$. The analysis between queue $\mathcal{DK}_2$ and queue $\mathcal{B}$ is similar to the control function of stage 2 in the Extended Kanban policy. The production rate out of stage 2 can be approximated as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2[s_2 - x_2] & ; x_1 + x_2 \geq s_2, x_2 \geq s_2 - k_2 \\ \mu_2[x_1] & ; x_1 + x_2 < s_2, x_2 \geq s_2 - k_2 \end{cases} \quad \text{(A.37)}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ; x_1 \geq k_2, x_2 < s_2 - k_2 \\ \mu_2(x_1) & ; x_1 < k_2, x_2 < s_2 - k_2 \end{cases} \quad \text{(A.38)}$$

Unlike the case for the Extended Kanban policy, the demand information authorizing the production of stage 1 in Generalized Kanban policy is independent of s_1 and k_1. As long as $x_2 \geq s_2 - k_2$, this demand formation will never be blocked. Similar analysis as in the case of Extended Kanban can be done and the control functions of stage 1 can be approximated as follow.

For $x_2 \geq s_2 - k_2$;

$$\mu_1(y_1) = \begin{cases} \mu_1[(s_1 + s_2) - (x_1 + x_2)] & ; (x_1 + x_2) \geq s_1 + s_2 - k_1 \\ \mu_1[k_1] & ; (x_1 + x_2) < s_1 + s_2 - k_1 \end{cases} \quad \text{(A.39)}$$

For $x_2 < s_2 - k_2$;

$$\mu_1(y_1) = \begin{cases} \mu_1(s_1 + k_2 - x_1) & ; x_1 \geq s_1 + k_2 - k_1 \\ \mu_1(k_1) & ; x_1 < s_1 + k_2 - k_1 \end{cases} \quad \text{(A.40)}$$

A 6 Control function of Extended CONWIP Kanban policy

The control function of stage 2 is similar to the ones in Extended Kanban and Generalized Kanban policies. The production rate out of stage 2 can be approximated as follow.

$$\mu_2(y_2) = \begin{cases} \mu_2[s_2 - x_2] & ; x_1 + x_2 \geq s_2, x_2 \geq s_2 - k_2 \\ \mu_2[x_1] & ; x_1 + x_2 < s_2, x_2 \geq s_2 - k_2 \end{cases} \tag{A.41}$$

$$\mu_2(y_2) = \begin{cases} \mu_2(k_2) & ; x_1 \geq k_2, x_2 < s_2 - k_2 \\ \mu_2(x_1) & ; x_1 < k_2, x_2 < s_2 - k_2 \end{cases} \tag{A.42}$$

To analyze the control function of stage 1, the first condition to be considered is when the order on-hold in queue $\mathcal{D}_1$ is positive. Here, we consider the condition on the amount of CONWIP instead of the amount of the sum of the kanbans as in the Extended Kanban case because the amount of CONWIP is generally less than the sum of the kanbans, i.e. $C \leq k_1 + k_2$, otherwise the Extended CONWIP Kanban policy will exactly be the same as the Extended Kanban policy. The order on-hold in queue $\mathcal{D}_1$ will always be positive if $x_2 \leq s_1 + s_2 - C$ and queue $\mathcal{D}_I$ can be left out of the analysis of the first synchronization station. The numbers of parts in the other two queues under the above condition can be computed as follow.

$$NQ(\mathcal{C}) = C\text{-}\, y_1 - x_1 \tag{A.43}$$

$$NQ(\mathcal{K}_1) = k_1 - y_1 - NQ(\mathcal{B})$$

$$= k_1 - y_1 - (x_1 - y_2) \tag{A.44}$$

$$(C - x_1 - y_1) \cdot (k_1 - x_1 + y_2 - y_1) = 0 \tag{A.45}$$

Using the fact that both queues are non-negative, the numbers of part being processed in stage 1 can be written as follow.

$$y_1 = \begin{cases} C - x_1 & ; x_1 \geq C - k_1, x_2 < s_1 + s_2 - C \\ k_1 - x_1 + y_2 & ; x_1 < C - k_1, x_2 < s_1 + s_2 - C \end{cases} \tag{A.46}$$

By substituting the value of y_2 from Equation (xxx) – (xxx) and eliminating some infeasible value, the production rate out of stage 1 can be approximated as follow.

For $x_2 < s_1 + s_2 - C$;

$$\mu_1(y_1) \quad = \quad \begin{cases} \mu_1(C - x_1) & ; x_1 \geq C - k_1 \\ \mu_1(k_1) & ; x_1 < C - k_1 \end{cases} \tag{A.47}$$

For the case when $x_2 \geq s_1 + s_2 - C$, the amount of free CONWIP is always positive and queue $\mathcal{C}$ can be left out of analysis. The numbers of parts in the other two queues in the first synchronization station can be computed as follow.

$$\begin{aligned} NQ(\mathcal{D}_1) &= (s_1 + s_2) - (y_1 + x_1 + x_2) \\ &= (s_1 + s_2) - (x_1 + x_2) - y_2 \end{aligned} \tag{A.48}$$

$$\begin{aligned} NQ(\mathcal{K}_1) &= k_1 - y_1 - NQ(\mathcal{B}) \\ &= k_1 - x_1 + y_2 - y_1 \end{aligned} \tag{A.49}$$

Using the property of the synchronization station and the fact that numbers of parts in each queue are non-negative, the number of parts in stage 1 can be expressed as follow.

$$y_1 \quad = \begin{cases} (s_1 + s_2) - (x_1 + x_2) & y_2 \geq (s_1 + s_2) - k_1 - x_2 \\ k_1 - x_1 + y_2 & y_2 < (s_1 + s_2) - k_1 - x_2 \end{cases} \tag{A.50}$$

By substituting the value of y_2 from Equation (xxx) – (xxx) and eliminating some infeasible values, the production rate out of stage 1 can be approximated as follow.

For $x_2 \geq s_1 + s_2 - C$;

$$\mu_1(y_1) \quad = \quad \begin{cases} \mu_1[(s_1 + s_2) - (x_1 + x_2)] & ; x_1 + x_2 \geq s_1 + s_2 - k_1 \\ \mu_1[k_1] & ; x_1 + x_2 < s_1 + s_2 - k_1 \end{cases} \tag{A.51}$$

APPENDIX B

PRODUCTION CAPACITY OF TWO-STAGE PRODUCTION SYSTEM

B 1 Base Stock control policy

Production Capacity $(x_2 \to -\infty)$:

$$\mu_1(x_1, x_2) \xrightarrow[x_2 \to -\infty]{} \mu_1(\infty)$$

$$\mu_2(x_1, x_2) \xrightarrow[x_2 \to -\infty]{} \mu_2(\infty)$$

$\therefore$ Production capacity does not depend on *s*, rather it's approaching the bottleneck rate of each stage.

B 2 Single-stage CONWIP and single-stage ECK control policies

Production Capacity $(x_2 \to -\infty)$:

$$\mu_1(x_1, x_2) \xrightarrow[x_2 \to -\infty]{} \mu_1(C - x_1)$$

$$\mu_2(x_1, x_2) \xrightarrow[x_2 \to -\infty]{} \mu_2(x_1)$$

$\therefore$ Production capacity depends on C

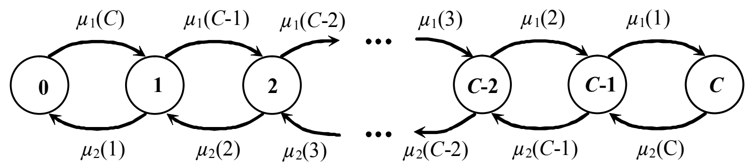


Figure B-1: Transition rate diagram of saturated CONWIP control system

From Figure B-1, the production capacity of single-stage CONWIP and single-stage ECK control policies can be calculated as follow:

For $i = 1,...,C$; $\quad P_i \quad = \quad \left[\prod_{j=1}^{i} \frac{\mu_1(C+1-j)}{\mu_2(j)}\right] \cdot P_0$ (B.1)

Solve for P_0 from; $\quad \sum_{i=0}^{C} P_i \quad = \quad 1$ (B.2)

Solve for each of P_i

Production Capacity; $\quad \mu_{\max} \quad = \quad \sum_{i=1}^{C} \{\mu_2(i) \cdot P_i\}$ (B.3)

B 3 Two-stage Kanban and two-stage Extended Kanban control policies

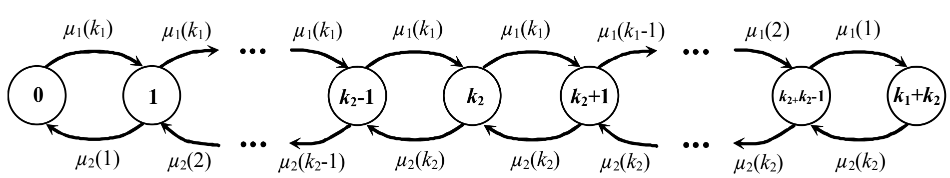


Figure B-2: Transition rate diagram of saturated two-stage Kanban control system

From Figure B-2, the production capacity of two-stage Kanban and two-stage Extended Kanban can be calculated as follow:

For $i = 1,\dots,k_2$; P_i $\quad = \quad \left[\dfrac{[\mu_1(k_1)]^i}{\prod_{j=1}^{i} \mu_2(j)} \right] \cdot P_0$ (B.4)

For $i = k_2+1,\dots,k_1+k_2$; $\quad P_i \quad = \quad \left[\dfrac{[\mu_1(k_1)]^{k_2} \cdot \prod_{j=1}^{i-k_2} \mu_1(k_1+1-j)}{\prod_{j=1}^{k_2} \mu_2(j) \cdot [\mu_2(k_2)]^{i-k_2}} \right] \cdot P_0$ (B.5)

Solve for P_0 from; $\quad \sum_{i=0}^{k_1+k} P_i \quad = \quad 1$ (B.6)

Solve for each of P_i

Production Capacity; $\quad \mu_{\max} \quad = \quad \sum_{i=1}^{k_2} \{\mu_2(i) \cdot P_i\} + \mu_2(k_2) \cdot \sum_{i=k_2+1}^{k_1+k_2} P_i$ (B.7)

B 4 Two-stage Generalized Kanban control policy

Case 1: Two-stage GK control system having $k_1 > s_1$

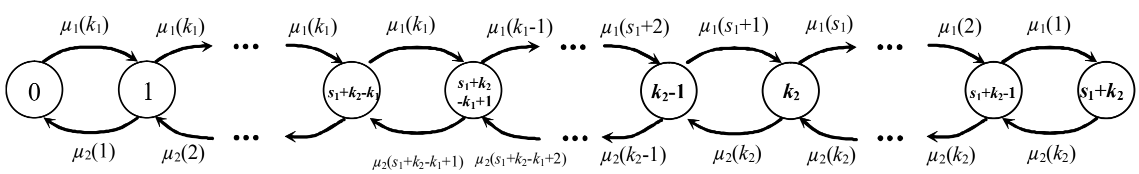


Figure B-3: Transition rate diagram of saturated two-stage GK control system $(k_1 > s_1)$

From Figure B-3, the production capacity of two-stage Generalized Kanban having $k_1 > s_1$ can be calculated as follow:

For $i = 1,\ldots,s_1 + k_2 - k_1$; $\quad P_i \quad = \quad \left[\dfrac{[\mu_1(k_1)]^i}{\prod_{j=1}^{i} \mu_2(j)} \right] \cdot P_0$ (B.8)

For $i = s_1 + k_2 - k_1 + 1,\ldots,k_2$; $P_i =$

$$\left[\frac{[\mu_1(k_1)]^{s_1+k_2-k_1} \cdot \prod_{j=1}^{i-(s_1+k_2-k_1)} \mu_1(k_1 + 1 - j)}{\prod_{j=1}^{i} \mu_2(j)} \right] \cdot P_0$$

(B.9)

For $i = k_2 + 1,\ldots,s_1 + k_2$; $P_i =$

$$\left[\frac{[\mu_1(k_1)]^{s_1+k_2-k_1} \cdot \prod_{j=1}^{i-(s_1+k_2-k_1)} \mu_1(k_1 + 1 - j)}{\prod_{j=1}^{k_2} \mu_2(j) \cdot [\mu_2(k_2)]^{i-k_2}} \right] \cdot P_0$$

(B.10)

Solve for P_0 from; $\quad \sum_{i-0}^{s_1+k_2} P_i = \quad 0$ (B.11)

Solve for P_i

Production Capacity; $\quad \mu_{\max} \quad = \quad \sum_{i=1}^{k_2} \{\mu_2(i) \cdot P_i\} + \mu_2(k_2) \cdot \sum_{i=k_2+1}^{s_1+k_2} P_i$ (B.12)

Case 2: Two-stage GK control system having $k_1 < s_1$

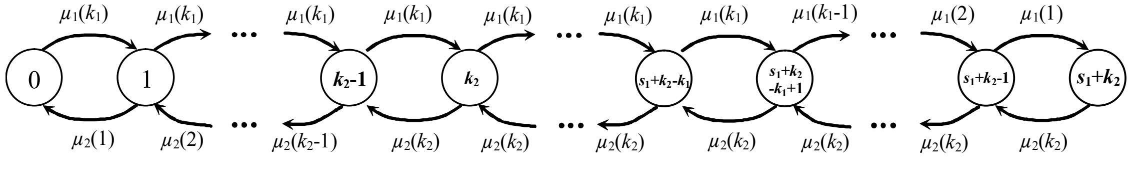


Figure B-4: Transition rate diagram of saturated two-stage GK control system $(k_1 < s_1)$

From Figure B-4, the production capacity of two-stage Generalized Kanban having $k_1 < s_1$ can be calculated as follow:

For $i = 1,...,k_2$; $\quad P_i \quad = \quad \left[\dfrac{[\mu_1(k_1)]^i}{\prod_{j=1}^{i}\mu_2(j)}\right]\cdot P_0$ (B.13)

For $i = k_2 + 1,...,s_1 + k_2 - k_1$; $P_i = \left[\dfrac{[\mu_1(k_1)]^i}{\prod_{j=1}^{k_2}\mu_2(j)\cdot[\mu_2(k_2)^{i-k_2}]}\right]\cdot P_0$ (B.14)

For $i = s_1 + k_2 - k_1 + 1,...,s_1 + k_2$;

$$P_i = \left[\frac{[\mu_1(k_1)]^{s_1+k_2-k_1}\cdot\prod_{j=1}^{i-(s_1+k_2-k_1)}\mu_1(k_1+1-j)}{\prod_{j=1}^{k_2}\mu_2(j)\cdot[\mu_2(k_2)]^{i-k_2}}\right]\cdot P_0 \quad \text{(B.15)}$$

Solve for P_0 from; $\quad \sum_{i-0}^{s_1+k_2} P_i = 0$ (B.16)

Solve for P_i

Production Capacity; $\quad \mu_{\max} \quad = \quad \sum_{i=1}^{k_2}\{\mu_2(i)\cdot P_i\} + \mu_2(k_2)\cdot\sum_{i=k_2+1}^{s_1+k_2} P_i$ (B.17)

B 5 Two-stage CONWIP Kanban and two-stage ECK control policies

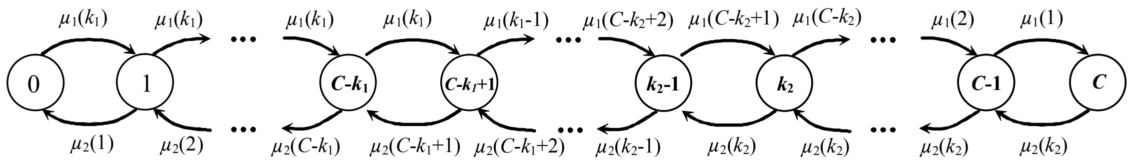


Figure B-5: Transition rate diagram of saturated two-stage CK control system

From Figure B-5, the production capacity of two-stage CONWIP Kanban and two-stage Extended CONWIP Kanban can be calculated as follow:

For $i = 1,\ldots,C-k_1$; $$P_i = \left[\frac{[\mu_1(k_1)]^i}{\prod_{j=1}^{i}\mu_2(j)}\right]\cdot P_0 \quad \text{(B.18)}$$

For $i = C-k_1+1,\ldots,k_2$; $$P_i = \left[\frac{[\mu_1(k_1)]^{C-k_1}\cdot\prod_{j=1}^{i-(C-k_1)}\mu_1(k_1+1-j)}{\prod_{j=1}^{i}\mu_2(j)}\right]\cdot P_0 \quad \text{(B.19)}$$

For $i = k_2+1,\ldots,C$; $$P_i = \left[\frac{[\mu_1(k_1)]^{C-k_1}\cdot\prod_{j=1}^{i-(C-k_1)}\mu_1(k_1+1-j)}{\prod_{j=1}^{k_2}\mu_2(j)\cdot[\mu_2(k_2)]^{i-k_2}}\right]\cdot P_0$$

(B.20)

Solve for P_0 from; $$\sum_{i-0}^{k_1+k_2}P_i = 0 \quad \text{(B.21)}$$

Solve for P_i

Production Capacity; $$\mu_{\max} = \sum_{i=1}^{k_2}\{\mu_2(i)\cdot P_i\} + \mu_2(k_2)\cdot\sum_{i=k_2+1}^{C}P_i \quad \text{(B.22)}$$

REFERENCES

Berkley, B. J. (1992). "A review of the kanban production control research literature." Production and Operations Management **1**(4): 393-411.

Bonvik, A.M. (1997). "C.E. Couch and S.B. Gershwin, Comparison of production-line control mechanisms" International Journal of Production Research **35**(3): 789-804.

Buzacott, J. A. (1989). "Queuing models of kanban and MRP controlled production systems." Engineering Cost and Production Economics **17**: 3-20.

Buzacott, J. A. and G. J. Shantikumar (1993). Stochastic Models of Manufacturing Systems. Englewood Cliffs, NJ, Prentice-Hall.

Clark, A. J. and H. Scarf (1960, Jul). "Optimal policies for the multi-echelon inventory problem." Management Science **6**(4): 475-490.

Dallery, Y. and G. Liberopoulos (1995). "A new kanban-type pull control mechanism for multi-stage manufacturing systems." Proceedings of the 3rd European Control Conference, Rome, Italy: 3543-3548.

Dallery, Y. and G. Liberopoulos (2000). "Extended kanban control system: combining kanban and base stock." IEEE Transactions **32**: 369-386.

Dallery, Y. and S. B. Gershwin (1992, Dec). "Manufacturing flow line systems: A review of models and analytical results." Queuing Systems Theory and Application **12**(1-2): 3-94. Special issue on queuing models of manufacturing systems.

Dallery, Y., Z. Liu, et al. (1997). "Properties of fork/join queuing networks with blocking under various operating mechanisms." IEEE Transactions on Robotics and Automation **13**(4): 503-518.

Deleersnyder, J. L., T. J. Hodgson, et al. (1989). "Kanban Controlled Pull Systems: An Analytical Approach." Management Science **35**: 1079-1091.

Frein, Y., M. D. Mascolo, et al. (1994). "On the design of generalized kanban control systems." To appear in International Journal of Operations and Production Management, special issue on Modelling and Analysis of Just-in-Time Manufacturing Systems.

Geraghty, J. and Heavey, C. (2004) "A Comparison of Hybrid Push/Pull and CONWIP/Pull production inventory control policies." International Journal of Production Economics **91**(1): 75-91.

Gershwin, S. B. (1994). Manufacturing Systems Engineering. Englewood Cliffs, NJ, Prentice-Hall.

Groenvelt, H. (1993). "The just-in-time system." Handbooks in Operations Research and Management Science **4**.

Groenvelt, H. (1993). The Just-in-Time System. S. C. G. e. a. E. Handbooks in OR & MS 4, Elsevier Science Publishers B.V., Amsterdam**:** 629-671.

Hall, R. W. (1983) "Zero Inventories." Dow Jones-Irwin, Homewood, IL.

Hodgson, T.J. and Wang D. (1991a) "Optimal hybrid push/pull control strategies for a parallel multi-stage system: Part I." International Journal of Production Research **29**(6): 1279-1287.

Hodgson, T.J. and Wang D. (1991b) "Optimal hybrid push/pull control strategies for a parallel multi-stage system: Part II." International Journal of Production Research **29**(7): 1453-1460.

Hopp, W. J. and M. L. Spearman (1996). Factory Physics. New York, NY, McGraw-Hill.

Hopp, W.J. and Roof, M.L. (1998) "Setting WIP levels with statistical throughput control (STC) in CONWIP production lines." International Journal of Production Research **36**(4): 867-882.

Karaesmen, F. and Y. Dallery (2000). "A performance comparison of pull type control mechanisms for multi-stage manufacturing." International Journal of Production Economics **68**: 59-71.

Kimball, G. (1988). "General principles of inventory control." Journal of Manufacturing and Operations Management **1**(1): 119-130.

Liberopoulos, G. and Y. Dallery (2000). "A unified framework for pull control mechanisms in multi-stage manufacturing systems." Annals of Operation Research **93**: 325-355.

Liberopoulos, G. and Y. Dallery (2002). "Base stock versus WIP cap in single-stage make-to-stock production-inventory systems." IEEE Transactions **34**: 627-636.

Monden, Y. (1983). "Toyota Production System: Practical Approach to Production Management." Industrial Engineering and Management Press.

Ohno, T. (1988) "Toyota Production System: Beyond Large Scale Production." Productivity Press, Cambridge, MA.

Philipoom, P. R., L. P. Rees, et al. (1987). "Dynamically Adjusting the Number of Kanbans in a Just-in-Time Production System Using Estimated Values of Lead Time." IEEE Transactions: 199-207.

Rees, L. P., P. R. Philipoom, et al. (1987). "Dynamically adjusting the number of kanbans in a just-in-time production system using estimated values of lead-time." IIE Transactions **19**(2): 199-207.

Ribio, R. and L. M. Wein (1996). "Setting Base Stock Levels Using Product-form Queueing Networks." Management Science **42**(2): 259-268.

Shingo, S. (1989). A Study of the Toyota Production System from an Industrial Engineering Viewpoint., Cambridge, MA: Productivity Press.

Spearman, M. L. (1991). "An Analytic Congestion Model for Closed Production Systems with IFR Processing Times." Management Science **37**(8): 1015-1029.

Spearman, M. L. and M. A. Zazanis (1992). "Push and Pull Production Systems: Issues and Comparisons." Operation Research **40**(3): 521-532.

Spearman, M. L., D. L. Woodruff, et al. (1990). "CONWIP : a pull alternative to kanban." International Journal of Production Research **28**(5): 879-894.

Takahashi K. and Nakamura N. (2002) "Comparing reactive Kanban and reactive CONWIP." Production Planning & Control **13**(8): 702-714.

Takahashi, K. (2003). "Comparing reactive Kanban systems." INT. J. PROD. RES. **41**(18): 4317-4337.

Takahashi, K. and Nakamura, N. (1999) "Reacting JIT ordering systems to the unstable changes in demand." International Journal of Production Research **37**(10): 2293-2313.

Tardif, V. and L. Maaseidvaag (1999). "An Adaptive Approach to Controlling Kanban Systems." European Journal of Operational Research.

Tardif, V. and Maaseidvaag, L. (2001) "An adaptive approach to controlling Kanban systems." European Journal of Operational Research **132**(2): 411-424.

Van Ryzin, G. (1987). Control of Manufacturing Systems with Delays. Master's Thesis, Massachusetts Institute of Technology.

Van Ryzin, G., S. X. C. Lou, et al. (1993, Sep). "Production control for a tandem two-machine system." IIE Transactions **25**(5): 5-20.

Veatch, M. H. and L. M. Wein (1994, Mar-Apr). "Optimal control of a two-station tandem production/inventory system." Operations Research **42**(2): 337-350.

Whitt, W. (1984) "Open and Closed models for networks of queues." AT&T Bell Lab. Tech. J. **63**: 1911-1978.

Zipkin, P. (1989). "A kanban-like production control system: analysis of simple models." Research Working Paper No. 89-1, Graduate School of Business, Columbia University, New York.

Zipkin, P. (1991). "Does manufacturing need a JIT revolution?" Harvard Business Review(January-February): 40-50.